

Better Kohonen Neural Network in Radar Images Compression

Tomasz Praczyk

Naval University, Śmidowicza 69, Gdynia, Poland

e-mail: T.Praczyk@amw.gdynia.pl

(Rec. 1 September 2005)

Abstract: Maritime radar image may be a basis for a perspective ship position fixing system. Images obtained from navigational radars possess usually great amount of information. Using all of these information is usually practically impossible. It is necessary to compress information imbedded in each image to the size acceptable from the practical point of view. Such an effect may be obtained by application of feature extraction methods. This article presents one of such methods that is based on capabilities of modified self-organizing Kohonen neural network.

Key words: maritime navigation, feature extraction, neural networks

1. INTRODUCTION

In recent years, positioning in navigation was dominated by satellite systems, especially by GPS. These systems are used at sea (maritime navigation), in the air (airplane navigation) as well as on the ground (land navigation). Basing solely on satellite systems carries, however, the risk of a sudden loss of navigational information in case of a sudden malfunction of such system, its power-down, inclusion of noise or coding data reaching the user. Therefore, there is necessity of being able to use one more, autonomous (independent of external information sources) system that allows automatic position fixing.

One of the methods, which offer such capabilities, is an application of sea radar and getting an image out of it. All bigger sea going vessels are equipped with such radar and its application in navigation is not a new idea. However, currently it is only a tool allowing a navigator obtaining information necessary to fix position in a classic way, which means by using crossing position lines determined for reference points of know position. It is quite common that it is difficult even for an experienced navigator to identify properly points, which could be used to determine our accurate position. In such case, methods based on comparison of a registered radar image with a model image from a database should be used to determine approximate ship's position [11].

New approach to the problem of determining ship's position at sea is using for that purpose artificial intelligence methods such as artificial neural networks. They may constitute the reasoning element which task is to assign to registered radar image corresponding position. Such func-

tion may be fulfilled for example by such neural networks as: multilayer perceptron with linear output neurons, RBFNN, HRBFNN, GRNN and fuzzy neural network.

The most important criterion for estimating a positioning system is its accuracy. Willing to obtain precise positioning system it is necessary to collect large number of radar images from a coastal area that we are interested in. This usually means hundreds and even thousands of images, a couple of images per position. It is a massive amount of data, which causes that we deal with a complex task. Learning a neural network basing on such a massive data may become practically impossible and, at least very troublesome and time consuming. Thus, it is necessary to apply different methods that simplify the complexity of the problem. One of the methods used is problem decomposition into simpler problems, which means that sea area is divided into smaller regions – each with its own positioning system. Another method consists in removing redundant radar images from the learning set. This way, in the learning process only essential images are used that increase the knowledge of the system [1]. The last method for reducing complexity is the application of feature extraction methods (leading to image compression). A condensing of information imbedded in radar image is applied to the degree acceptable from the point of view of time of further calculations.

Feature extraction methods are not used solely as a panacea for acceleration of calculations either in the inference phase or in the learning one. Very often, their application is a necessary condition for a proper functioning of the positioning system. Such situation exists for example when learning data that are at our disposal are far different

from images that are subjected for positioning (learning images may be generated from an electronic or paper chart, positioning is used basing on radar images). In such case all methods mentioned above without input data pre-processing, without bringing all that data into a common denominator are unable to perform their task. Then, feature extraction methods are useful because they get from images only information, which are essential for the task, omitting everything that is unnecessary or even detrimental.

Ship positioning system

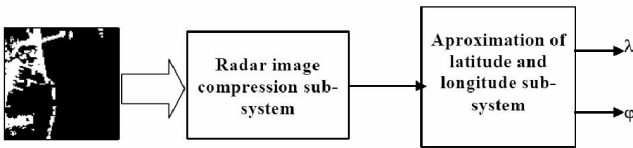


Fig. 1. An outline of the positioning system

One of the solutions that allow feature extraction from radar images is self-organizing Kohonen neural network. This neural network usually consists of one layer of m neurons. An input of each neuron constitutes whole input vector $\mathbf{u} = [u_1, u_2, \dots, u_k]$ (segment of radar image). The task of each neuron is to calculate distance between the input vector \mathbf{u} and a weights vector $\mathbf{W}_i = [W_{i1}, W_{i2}, \dots, W_{ik}]$ (where i means number of neuron). On the output of the neural network usually only one neuron is active – neuron w fulfilling below equation [5]:

$$d(\mathbf{u}, \mathbf{W}_w) = \min_{1 \leq i \leq m} d(\mathbf{u}, \mathbf{W}_i) \quad (1)$$

where $d(\mathbf{u}, \mathbf{W})$ means distance between the vector \mathbf{u} and vector \mathbf{W} . Kohonen neural network has a potential for cumulating medium features of vectors from the learning set. A learned network returns an index of a neuron, which is the closest to a vector presented to the network. It means that Kohonen neural network conducts a sort of coding by assigning a number resulting from the organization of the network – the index of one of its neurons, to an input vector (image segment). Application of classic Kohonen network assumes that only one, the nearest neuron at the output of the network is taken into account. In order to obtain more complete information about the position of the input vector (image segment presented at the input of the network) in the input domain, it is possible to apply a network with a greater number of outputs taken into account. The outputs of such a network indicate those neurons which are the most similar to the input vector. We can implement it by arranging neurons of the network in accordance with a distance of each of them to the actual input vector $d_0 < d_1 < d_2 < \dots < d_{m-1}$ where $d_s = d(\mathbf{u}, \mathbf{W}_{s(i)})$ denotes

a distance of neuron i , which has an index s in ordered sequence of neurons, to the input vector \mathbf{u} . At the beginning of the sequence of arranged neurons are the neurons, which are the closest to the input vector. The farthest neurons occur at the end of this sequence. Active neurons, in such case are neurons, which are on the same beginning of the sequence.

When building the maritime coastal spare positioning system and deciding to use in a compression phase Kohonen neural network it is necessary to determine: division of radar images into segments (number of segments, their size and the amount of information that is received from the neural network) and the number of neurons. Assuming that the compressed image size (number of features) is fixed, it is necessary to adjust to it the size of the segment (it is assumed that segments have the same size) and the number of outputs of neural network taken into account. If the number of all segments (N_{segments}) is equal to the size of the image that we want to obtain at the output of the compression subsystem ($N_{\text{compressed}}$), we can take into consideration only that output neuron which is the closest to the image presented at the input to the network ($N_{\text{outputs}} = 1$). However, if we decide on bigger segments, then in such case we should take into account a greater number of neurons ($N_{\text{outputs}} > 1$).

There are many image compression methods but there is a problem with estimating their usefulness in maritime positioning systems. Results presented in [8] and [9] are based on estimating the compressing Kohonen neural network using position accuracy achieved by the positioning system based on radar images compressed by this neural network. Thus, we have the estimation of the given solution without the knowledge of contribution of each part of the system to the final result.

For the purpose of estimating the solution used in the compression subsystem, an error function was defined, minimum of which determines the configuration of this subsystem, which faithfully preserves all relations existing between original radar images – radar images similar to each other have representatives in the domain of compressed images also similar to each other. Solutions, which perform a dispersion of the compressed equivalents of radar images from close positions, are treated as disadvantageous for the position approximation system being built. They would increase the speed of changeability of the approximated function in areas where exist pieces of data similar to each other but with different values of position function – $f(\text{comp_i}) = \text{position}$ where comp_i is an compressed radar image and position is a latitude and longitude vector. In order to secure a proper precision of the positioning system, these areas had to be represented by a greater number of learning data, prolonging this way the computation time in the learning and reasoning phase. The radar image compression sub-system estimation function is as follows [6]:

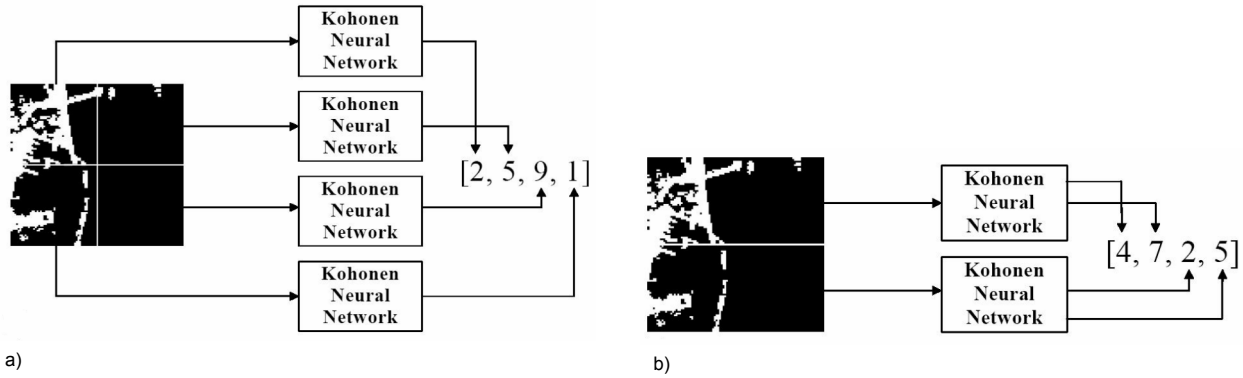


Fig. 2. The connection between the size of input data, the number of segments and the number of used network outputs

$$E = \frac{1}{c} \sum_{i < j}^n \left[(a_{ij} - a_{ij}^*)^2 / a_{ij} \right] \quad (2)$$

$$c = \frac{n(n-1)}{2} \quad (3)$$

where: n – the number of test radar images;
 i, j – indexes of consecutive radar images and their compressed equivalents;
 a_{ij} – normalized Euclidean distance between two radar images;
 a_{ij}^* – normalized Euclidean distance between two compressed images.

In order to verify the possibility of Kohonen neural network application as a radar images compression subsystem in maritime spare coastal positioning system experiments were conducted. Results of these experiments include the main part of the article. Section 2 illustrates classical network usage and accomplished results. In section 3, a modification of Kohonen neural network is presented and results of its application to radar images compression are reported. Concluding remarks are given in section 4.

2. COMPRESSION ON THE BASIS OF CLASSICAL KOHONEN NEURAL NETWORK

Conducted experiments consisted in examination of the influence of the neurons number in Kohonen neural network and value of N_{segments} and N_{outputs} at fixed $N_{\text{compressed}}$ on the compressed radar image. For that purpose, 31 original gray scale radar images coming from the Gdansk Bay area (distance between consecutive registrations of radar images is about 600 m) as well as 93 derivatives of these images were used. Each original image had 3 additional equivalents (converted from it), which sums to 4 image series – each consisting of 31 images from different positions (primary series no. 1 with the originals and series no. 2, 3 and

4 with the copies). Images with the same indexes in each of the series corresponded to the same ship's position (position registered using GPS). Additional radar images were constructed by rotation of original images at the range $\langle -3^\circ, +3^\circ \rangle$. The rotation was used in order to consider a gyrocompass error. Gyrocompass is a ship device, which serves to arrange radar images according to the N–S direction. After the rotation, deformations were introduced to rotated images. The magnitude of introduced deformations was different for each images series. The smallest differences occurred between series No. 1 and No. 2, next between series No. 1 and No. 3 and the largest disparity was between images series no. 1 and no. 4.

All images were reduced to the size of 100×100 pixels and then they were subjected to the process of segmentation. The research was divided into three phases:

- First phase – $N_{\text{compressed}} = 200$, $N_{\text{segments}}^1 = 200$ and $N_{\text{outputs}}^1 = 1$
- Second phase – $N_{\text{compressed}} = 200$, $N_{\text{segments}}^2 = 100$ and $N_{\text{outputs}}^2 = 2$
- Third phase – $N_{\text{compressed}} = 200$, $N_{\text{segments}}^3 = 25$ and $N_{\text{outputs}}^3 = 8$

In each of the three phases, 20 different Kohonen network configurations were tested – from 10 neuron network to the network consisting of 250 output neurons. For each configuration, four copies of the network were built. Every copy was prepared on the basis of different learning set and learning algorithm (first copy – neural gas algorithm and primary series no. 1, second copy – neural gas algorithm and image series no. 2, third copy – SCS (Soft Competition Scheme) algorithm and primary series no. 1, fourth copy – SCS algorithm and image series no. 2). During the experiments two learning algorithms were used to learn Kohonen neural network – neural gas algorithm and SCS algorithm. The neural gas algorithm proposed by Martinez, Berkovich and Schulten consists in arranging neurons of Kohonen network in accordance with distance of each of them to actual learning image. At the beginning of the sequence of

arranged neurons the closest neurons to the input image are. At the end of this sequence the farthest neurons occur. Then, the following procedure of updating weights of neurons is applied [5]:

$$\mathbf{W}_i(k+1) = \mathbf{W}_i(k) + h_i(k)G_i(\mathbf{u}, k)[\mathbf{u} - \mathbf{W}_i(k)] \quad (4)$$

$$G_i(\mathbf{u}, k) = e^{-\frac{s(i)}{l(k)}} \quad (5)$$

$$\eta_i(k) = \eta_i(0) \left(\frac{\eta_{\min}}{\eta_i(0)} \right)^{k/k_{\max}} \quad (6)$$

$$\lambda(k) = \lambda_{\max} \left(\frac{\lambda_{\min}}{\lambda_{\max}} \right)^{k/k_{\max}} \quad (7)$$

where:

- i – index of neuron of Kohonen neural network;
- k_{\max} – maximal number of epochs applied during learning process;
- η_{\min} – minimal value of η corresponding to $k = k_{\max}$;
- $\eta_i(0)$ – start value of η ;
- $\lambda_{\min}, \lambda_{\max}$ – minimal and maximal value of λ coefficient.

In case of SCS algorithm, the adaptation of weights of neurons proceeds according to the same formula (4) as in neural gas algorithm. The differences between algorithms occur in relation to $G_i(\mathbf{u}, k)$ and $\eta_i(k)$ [5]:

$$G_i(\mathbf{u}, k) = \frac{\exp\left(-\frac{\|\mathbf{u} - \mathbf{W}_i\|^2}{T(k)}\right)}{\sum_{j=1}^m \exp\left(-\frac{\|\mathbf{u} - \mathbf{W}_j\|^2}{T(k)}\right)} \quad (8)$$

$$\eta_i(k) = \frac{1}{1 + \sum_{j=1}^{k-1} G_i(\mathbf{u}, j)} \quad (9)$$

where

$$T(k) = T_{\max} \left(\frac{T_{\min}}{T_{\max}} \right)^{k/k_{\max}} \quad (10)$$

is the “temperature” parameter in k step.

During learning, the following values of algorithms parameters were applied – $\lambda_{\max} = 20$, $\lambda_{\min} = 0.001$, $\eta_i(0) = 0.8$, $\eta_{\min} = 0.001$, $T_{\max} = 1$, $T_{\min} = 0.00001$. The learning was finished always after reaching 10 000 learning epoch – $k_{\max} = 10\,000$. The number of entries of Kohonen network was equal to the size of a single segment – 50, 100 and 400 inputs. Thus, the learning set consisted of 6200 ($31 * N^1_{\text{segments}} = 6200$), 3100 ($31 * N^2_{\text{segments}} = 3100$) or 775 ($31 * N^3_{\text{segments}} = 775$) images – segments of original images. Radar images, independently of the phase, were compressed,

always to the size of 200 information units – $N_{\text{compressed}} = 200$.

Each of the trained Kohonen networks was used to compression of images coming from testing series no. 3 and 4 and learning series no. 1 and 2. Obtained feature vectors were used to perform assessment of the neural networks. Obtained results are presented on Fig. 3.

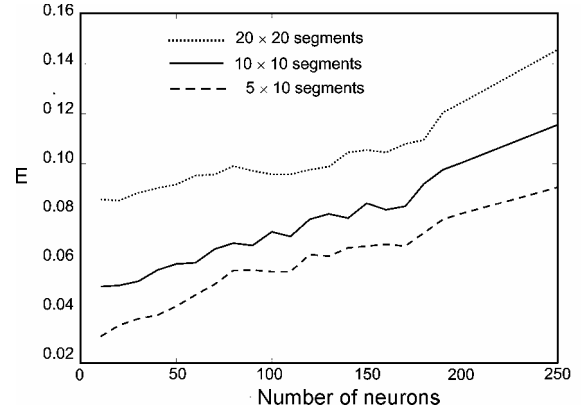


Fig. 3. Average results for all four compressing networks built for every network configuration and size of segment applied (20 * 20 segments – eight output neural network, segments sized 20 * 20 pixels, 10 * 10 segment – two output neural network, segments sized 10 * 10 pixels, 5 * 10 segments – one output neural network, segments sized 5 * 10 pixels)

For Kohonen neural network consisting only of one neuron, all input radar images are treated as the same. All of them possess the same counterparts in the feature vectors domain. If we assume, that:

$$a_{ij}^* = \frac{\sqrt{\sum_l^{r_{ij}} \Delta_l^{ij}}}{a_{\max}^*} \quad (11)$$

where r_{ij} – the number of different components of i^{th} and j^{th} feature vector (r_{ij} value reflects similarity between radar images – higher r_{ij} value means larger differences between images, lower r_{ij} informs about greater similarity between images), Δ_l^{ij} – random value corresponding to disparity between l^{th} elements of i^{th} and j^{th} feature vector, than in the situation with only one output neuron – $r_{ij} = 0$, $a_{ij}^* = 0$, for each i, j , $i < j$, and as a consequence $E = (1/c) \sum_{i < j} a_{ij}^*$. In our case $E = 0.6588$. Gradual increase in the number of neurons causes a decrease of these neurons responsibility areas, and segments of radar images less often find the way to influence zones of the same neurons. Therefore, the expansion of the network structure entails the increase of the r_{ij} value. The increase of \bar{r} value (denotes value fixed for all feature vectors pairs) as the network expands, generally causes also a rising tendency of random value \bar{a}^* . Initially, growth of \bar{a}^* results in decrease of error E . It reaches the minimal value for 10 neuron network, for which difference $\bar{a} - \bar{a}^*$ reached the smallest value among

tested Kohonen neural network structures. The addition of extra neurons to the network causes further growth of \bar{r} and \bar{a}^* , what causes increase of error E .

Δ_l^{ij} determines similarity between segments at index l coming from images i and j . The situation, in which $\Delta_l^{ij} = 0$, means that the network interpreted segments from both images as similar and assigned them the same neuron. The case where $\Delta_l^{ij} \neq 0$, means that the network classified segments from both images as different. In this situation, Δ_l^{ij} does not reflect differences between segments. It is a random value resulting from neurons arrangement and from random indexes assigned to each of them. Therefore, the only useful information, when we compare two feature vectors generated by Kohonen neural network, is r_{ij} value determining the number of nonzero elements of comprised vectors. The way of generating feature vectors representing radar images of the coast by Kohonen neural network causes that this information should be taken into account during building the final fixing position sub-system based on data prepared by classical Kohonen neural network. In case when the final inference sub-system will be implemented as one of radial neural nets (GRNN, RBFNN, HRBFNN), Euclidean distance – the most often used in neurons of such nets, should be replaced with distance (12).

$$d(\mathbf{y}^i, \mathbf{y}^j) = \sum_l^{N_{\text{compressed}}} \delta_l^{ij} = r_{ij} \quad (12)$$

$$\delta_l^{ij} = \begin{cases} 1 & \text{dla } \Delta_l^{ij} \neq 0 \\ 0 & \text{dla } \Delta_l^{ij} = 0 \end{cases} \quad (13)$$

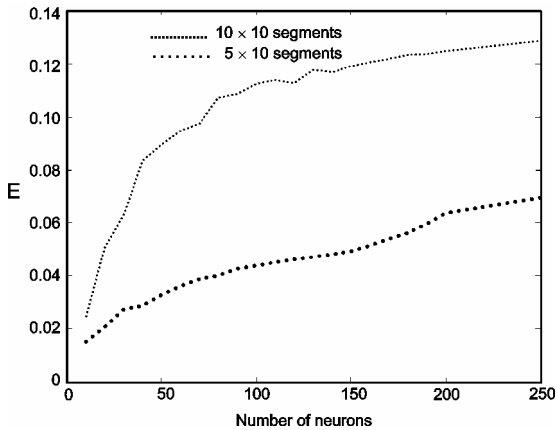


Fig. 4. Error E for compressing networks learned with neural gas algorithm depending on the number of neurons – a_{ij}^* is fixed according to the formula (12)

Worse results of many outputs and bigger segments solutions in comparison with one output and smaller segments conception result generally from quicker increase of r_{ij} value for the case with larger segments. It is much more

difficult to obtain the same feature vectors for two different larger segments (solution with more network outputs) than for corresponding to them collections of smaller segments (one output solution).

3. COMPRESSION ON THE BASIS OF MODIFIED KOHONEN NEURAL NETWORK

Results of classical Kohonen neural network application to radar images compression that have been presented so far show that classical network equipped with large quantity of neurons is not able to save in generated data similarities occurring between original radar images, generating higher and higher error (2) for larger and larger structures of the network.

Large quantity of neurons causes that division of images space into influences zones of individual neurons becomes thicker. The more neurons the thicker partition and higher accuracy of the compression – less information is lost during compression. Unfortunately, random selection of neuron's indexes causes that large-structure networks disperse concentrations of images originally located not far from each other. Close-positioned compressed images could move away from each other, in turn data registered for distant positions could be located after the compression close to each other contributing to undulation of the approximated position function. An application of the bigger network in the compression phase would prolong the inference time in the subsequent stage of calculations, because willing to ensure appropriate accuracy of the system, it would be necessary to extend training set that we use to learn the approximation position subsystem. An arrangement of neurons in such a way that the output signal of every neuron is adequate to the place, which it occupies in the images space, could be a solution of that problem. Sammon transformation best suits this task. It makes possible to project multidimensional vectors into smaller spaces. This way, the compressing network does not return a random index of neuron, but two-, three-dimensional vector (during conducted experiments we applied two-dimensional vector) that corresponds to neuron location in the images space. This is accomplished through minimization of the error defined below

$$E = \frac{1}{e} \sum_{i < j}^k \left[\frac{(d_{ij} - d_{ij}^+)^2}{d_{ij}^+} \right] \quad (14)$$

$$e = \sum_{i < j}^k d_{ij}^+ \quad (15)$$

- where: k – the number of neurons in Kohonen network;
- i, j – indexes of consecutive neurons and their two-dimensional equivalents;
- d_{ij} – Euclidean distance between two neurons;
- d_{ij}^+ – Euclidean distance between two neurons equivalents in the 2 dimensional space.

In minimization of (14) Sammon used Newton optimization method, simplified to the form:

$$y_{(pq)}(m+1) = y_{pq}(m) - \eta \Delta_{pq}(m) \tag{16}$$

where

$$\Delta_{pq}(m) = \frac{\frac{\partial E}{\partial y_{pq}}}{\sqrt{\frac{\partial^2 E}{\partial y_{pq}^2}}} \tag{17}$$

y_{pq} denotes q^{th} component of p^{th} neuron equivalent, whereas m denotes the next algorithm iteration and η is the learning rate.

Figure 5 depicts the proposed structure of Kohonen network modification. The classical solution is supplemented with an additional layer with memorized output vectors of Kohonen network neurons. Therefore, the proposed modified network is composed of two layers. Classical Kohonen neural network constitutes its first, input layer and the layer with neurons counterparts is a second, output layer of the network. Preparation of such construction consists in learning the classical network, for example with the application of SCS or neuron gas algorithm and then in creating the layer of output vectors on the basis of previously fixed neurons.

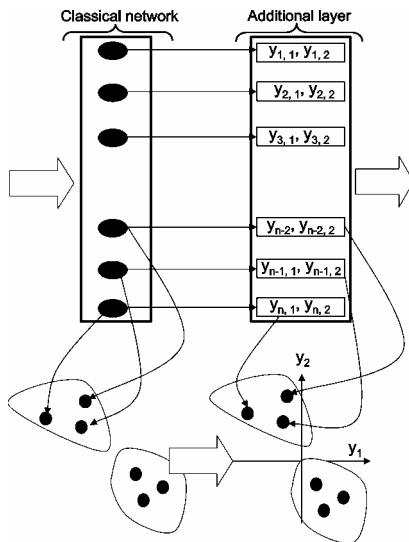


Fig. 5. The architecture of the proposed modification of classical Kohonen neural network

In order to verify the presented idea of radar images compression some tests were carried out. During the experiments, every modified compressing network was built on the basis of one of two output neural networks created previously. Therefore, in order to obtain complete compressing network it was necessary merely to determine an output signal of every neuron by means of Sammon trans-

formation. It was assumed that every neuron is represented by two-dimensional vector. This way, compressed images of the size of 200 units of information were obtained – one neuron associated with two-dimensional vector corresponding to the single segment of radar image.

Each modified network was used for compression of all 124 images that were at our disposal. Obtained feature vectors were used to perform assessment of every construction. Obtained averaged results are presented below.

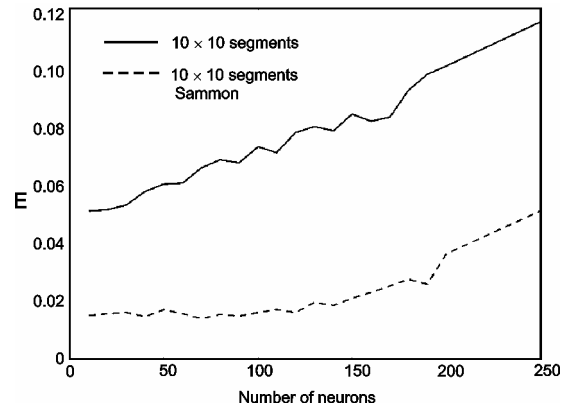


Fig. 6. Average results for all four modifications of Kohonen network and their prototypes built for every network configuration

For comparison, besides the analyzed solution, the above graph depicts also results of the application of classical networks that constituted the element of every modified network. Conducted experiments demonstrate that modified Kohonen neural network outperforms classical one, two and eight output Kohonen network, regardless of amount of neurons applied. Increase in the number of neurons in case of modification of Kohonen network induces the same effect as in case of conventional variety, *i.e.* increase of error (2). Such situation may result from gradually larger and larger difficulties in finding appropriately “good” output vectors for corresponding neurons when the number of optimized parameters increases. Such difficulties caused that neurons possessed worse and worse matched output values and consequently an effect of the network modification was less and less apparent.

Due to the fact that the main area of application of the methods presented in the paper is compression of radar images in spare, coastal, maritime positioning system, the test of the whole system, not only its single elements, would be also legitimate. Certainly, it is necessary to remember that results of such test will reflect performance of the whole solution, and determination of the compression subsystem contribution to the final outcome will be difficult in such case. Although, an application of such test to the compression sub-system evaluation seems to be rather questionable, it can answer a very important question,

namely – what accuracy of the positioning system can we obtain? However, in order to get reliable answer to this question we should have appropriately large set of high resolution radar images coming from larger sea area. Unfortunately, we do not have such set. Results of the tests with the application of images that we possess can be solely a source of rough information about abilities of the potential positioning system and its individual parts. Therefore, in order to obtain more reliable data concerning accuracy of the system, further research is necessary.

The estimated evaluation of usefulness of classical Kohonen network and its modification as the compressing element in spare, coastal, maritime positioning system can be obtained by presenting sample results of both solutions application in a prototype positioning system containing General Regression Neural Network as a reasoning (positioning) element. In order to compare both approaches, positioning GRNN was built, once on the basis of feature vectors prepared by classical two-output Kohonen network, and next time on the basis of patterns generated by two-output Kohonen network modification. Both compressing networks were learned with the application of basic, original learning set. Feature vectors corresponding to images coming from the same set constituted the learning set for the positioning network. Therefore, it included 31 radial neurons with memorized compressed learning radar images. Images from series no.3 were used to tune a shape of GRNN radial functions. Remaining images from series no.2 and 4 were used to test the positioning system.

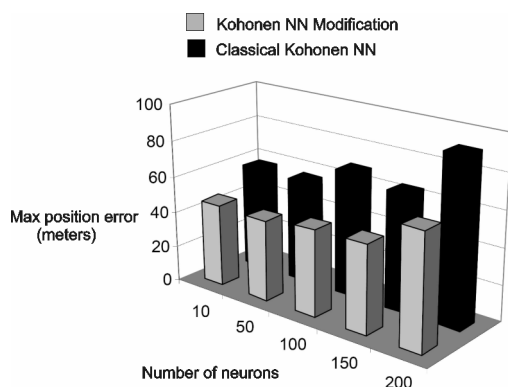


Fig. 7. Maximum position error for positioning systems built on the basis of classical Kohonen neural network and its modification

Presented results show that position accuracy in both cases is more or less the same. In general, we obtained better accuracy for smaller structures of compression networks what confirms correctness of the previous results. It is very difficult to refer to the same accuracy of the system since, as it was stated previously, data used during tests are rather inappropriate to this purpose. Even so, it is worth noticing that achieved accuracies are not so bad, given poor images quality, which rather imperfectly reflect a situation

at sea, and the fact that distances between individual learning images amounted to approximately 600 meters.

4. CONCLUSIONS

Building the coastal positioning system based on original radar images of a coastline, for the sake of numerical complexity of the fixing position process, we decided to precede it by the feature extraction from radar images, so that the inference was not based on the crude images usage but rather on their more compact counterparts application. One of the feature extraction methods that we decided to apply was self organizing Kohonen neural network. As a result of conducted research it turned out that the network, independently from learning algorithm, preserves concentrations in the best way for structures with not large number of neurons. The research demonstrated also that the one-output network solutions with usage of smaller segments achieve generally better outcomes then conceptions based on greater quantity of network outputs and using larger segments.

The experiments exhibited also that more complex structures of classical Kohonen network, despite large accuracy of compression (low losslessness degree), do not preserve concentrations that are apparent in the original radar images space. In the article the method allowing elimination of this shortcoming was presented – the modification of classical self-organizing Kohonen neural network. For fewer amount of neurons it allows reduction of error (2) in relation to the classical solution. Unfortunately, for structures that are more elaborate it behaves similarly as the network without modification. Such effect results from difficulties in finding vectors that constitute the network output, and consequently from the fact that the structure of the network applied during compression is far from optimal.

References

- [1] Y. Freund, H. Seung, E. Shamir and N. Tishby, *Selective sampling using the query by committee algorithm*, Machine Learning, **28**, 133-168 (1997).
- [2] B. Fritzke, *A growing neural gas network learns topologies*, *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, (1995).
- [3] J. Korbicz, A. Obuchowicz, D. Uciński, *Artificial neural networks*, AOW PLJ, Warsaw (1994) (In Polish).
- [4] G. Kuchariew, *Processing and analysis of digital images*, Szczecin Univ. of Technology, Szczecin (1999) (In Polish).
- [5] S. Osowski, *Neural networks in algorithmic sense*, WNT, Warsaw (1996) (In Polish).
- [6] T. Praczyk, *Radar images compression for the need of a positioning coastal system and an assessment of this process*, Annual of Navigation, no. 8, Gdynia (2004).
- [7] T. Praczyk, *The method of extraction of characteristic points from the radar image of the sea shore for the needs of positioning system*, Annual of Navigation, no. 8, Gdynia (2004).

- [8] T. Praczyk, *Kohonen neural network in radar images compression*, Scientific Bulletin no. 1, 2003, Naval University of Gdynia, Gdynia (2003).
- [9] T. Praczyk, *GRNN in radar images compression*, Scientific Bulletin no. 3, 2003, Naval University of Gdynia, Gdynia (2003).
- [10] W. Skarbek, *Methods of digital images representation*, AOW PLJ, Warsaw (1993) (In Polish).
- [11] A. Stateczny, *Comparative navigation*, Gdańsk Scientific Society, Gdańsk (2001) (In Polish).
- [12] A. Stateczny and T. Praczyk, *Artificial neural networks in radar image compression*, International Radar Symposium IRS, 2003, Drezno (2003).
- [13] R. Tadeusiewicz and M. Flesiński, *Imane recognition*, PWN, Warszawa (1991) (In Polish).



TOMASZ PRACZYK. *Education:* Military University of Technology, Warsaw – MSc (1996); Maritime University, Szczecin – PhD (2001). *Activities:* intelligent navigational systems, neural networks, genetic algorithms, neuroevolution, evolutionary reinforcement learning.