

# Analysis of the Modular Topology of Hybrid Neural Networks

O.I. Chumachenko

Technical Cybernetic Department  
National Technical University of  
Ukraine "Ihor Sikorsky Kyiv  
Polytechnic Institute"  
Kyiv, Ukraine  
chumachenko@tk.kpi.ua  
ORCID 0000-0003-3006-7460

K.D. Riazanovskiy

Technical Cybernetic Department  
National Technical University of  
Ukraine "Ihor Sikorsky Kyiv  
Polytechnic Institute"  
Kyiv, Ukraine  
kudgey@mail.ru

A.T. Kot

Technical Cybernetic Department  
National Technical University of  
Ukraine "Ihor Sikorsky Kyiv  
Polytechnic Institute"  
Kyiv, Ukraine  
anatoly.kot@gmail.com

**Abstract**—the report discusses the structures of modules composition and the problems associated with their learning. The optimal algorithm for modules learning for the classification problem is considered. Examples of specific structures are given. The structural-parametric synthesis of an ensemble of modules of neural networks are described. The results of the training of the modules and ensembles are presented, as well as a comparison with the results of the training of individual neural networks.

**Анотація**—у доповіді досліджується структура побудови модулів та проблеми, пов'язані з їх навчанням. Розглянуто оптимальний алгоритм навчання модулів для задачі класифікації. Наведені приклади конкретних структур. Розглянуто структурно-параметричний синтез ансамбля модулів нейронних мереж. Представлені результати навчання модулів і ансамблю модулів нейронних мереж, а також порівняння з результатами навчання окремих нейронних мереж.

**Keywords**—neural network; module; ensemble; topology; classification.

**Ключові слова**—нейронна мережа; модуль; ансамбль; топологія; класифікація.

## I. INTRODUCTION

The classification problem is one of the most frequent tasks that arises in the field of machine learning. It is common in many areas of life. Researchers from around the world are developing tools and algorithms to solve this problem [1][2][3]. Due to its diversity, the data classification problem cannot always be solved by the same tools and algorithms, one of the most successful are neural networks (NN) [4]. Due to their flexibility and scalability, they can solve the most diverse and complex problems that are beyond the power of classical machine learning algorithms.

For more difficult tasks, the new development branch of NN has become to combine them into one module: one large network, which consists of several base networks. In the future, to increase accuracy, it is possible to combine such modules into ensembles. This will compensate for the

shortcomings of each module by others, which will certainly have a positive impact on the final result of learning.

## II. PROBLEM STATEMENT

The goal of this report is to research the possible structures of the module, the topologies of the networks included in the module and the machine learning of the module for solving the classification problem. The further task of using the NN modules is to combine them into an ensemble.

## III. PROBLEM SOLUTION

### A. Modules

The module topology involves the sequential combination of several different neural network architectures. In general, the module operates in the same way as an individual NN. Its advantage is the combination of various data transformations, which allows to obtain more accurate results. The module topology is presented in Fig. 1.

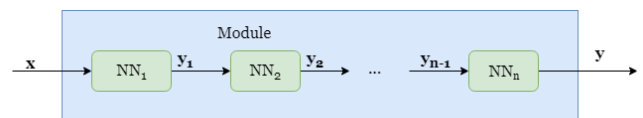


Fig. 1. Module topology

Modern technologies allow to operate with huge networks as with simple elements for building something larger, like LEGO bricks. The abstraction level of modern software is very high, everything is at an intuitive and understandable level, so the practical implementation of NN modules is a fairly easy task.

The main problem of modules composition is their learning algorithm. Inside the module there are several networks, therefore, we can train all these networks in different ways:

- all networks are trained together;

- some networks are trained together, some are trained separately;
- each network is trained separately on the training sets; then they are combined into a module.

The simplest way to train networks in a module is to use a genetic algorithm [5]. Each network has a certain number of its parameters. For each parameter a certain number of bits is allocated, then the parameters of all networks are combined into one chromosome in their bit representation. After this, the genetic algorithm is actually performed:

- 1) *generate initial population;*
- 2) *compute fitness;*
- 3) *selection;*
- 4) *crossover;*
- 5) *mutation;*
- 6) *compute fitness;*
- 7) *if population converged, then stop, else go to 3.*

As you can see, this training algorithm belongs to the first type: all networks are trained together. It has several disadvantages. Consider, for example, a module of three networks, each of which has 100 parameters. For each parameter, we will allocate 4 bits. The chromosome will contain  $3 * 100 * 4 = 1200$  genes. The convergence of this algorithm will require a tremendous amount of time and resources, so, in this case it is inefficient.

To maintain a balance of learning speed and accuracy, this report proposes using the network that is based on unsupervised learning as the first network. Its output goes to the input of the base network, which is trained separately under supervised learning. After the base network another network can be placed to refine the result.

The advantage of the structure presented above is that the first network is trained without a teacher very quickly, compared to the large base networks. It performs preprocessing (clustering, dimensionality reduction) of the input data, which ultimately has a positive effect on the following base networks. This preprocessing reduces the number of layers and neurons in the layers of the base network, so that it will be trained much faster and more accurately.

In some simple cases, a network that is based on unsupervised learning will already produce fairly accurate results, so the subsequent small base network will only refine them. In sum, the training of such two small networks will be much less than the training of one large base network.

In this report, the use of the Kohonen network [6] as the first network is proposed for solving the classification problem. It performs the separation of input data into groups, and the subsequent base network can determine the correct class label using the "hint" of the first network. Kohonen network is trained very quickly. It will reduce the dimension of the input data, and also determine the group of each input sample. In this case, samples that belong to the same class fall into one or neighboring groups. For more accurate work of

the Kohonen network, the use of an interpolation algorithm is required during training.

As a base network, various networks can be used (e.g. perceptron, radial-basis function network, NEFClassM, etc.).

The full topology of the proposed module is presented in Fig. 2.

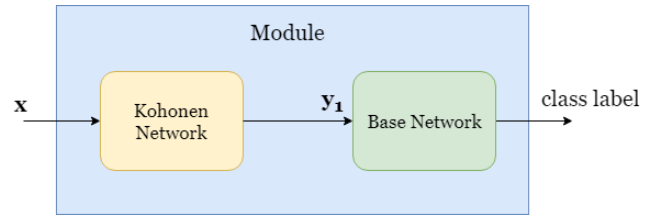


Fig. 2. Proposed module topology

To refine the classification result, bidirectional associative memory [8] can be used after the base network.

### B. Ensembles

In simple cases, the construction of a single module may be sufficient to achieve the required accuracy, but to solve complex problems it is necessary to use several modules combined into one ensemble.

The construction of the ensemble allows you to look at the problem from points of view of different modules. The usage of elements of the ensemble as modules presented in Fig. 2 instead of simple neural networks has a set of advantages. It requires less memory, takes less time to train and, as will be shown in the next part, has greater accuracy than an ensemble consisting of individual neural networks.

## IV. EXAMPLE ON REAL SET

For the experiment, modules were used, which consist of two networks. The first is the Kohonen network and the second is the base network. As the base networks were used: perceptron, radial-basis function network, counter propagation network, probabilistic neural network, NEFClassM, Naïve Bayes Classifier. In order to show a clear advantage of modular topology over individual NN, a comparison was made with the results obtained in [7]. For the experiment, the same Wine data set was used: 178 samples, 13 features represented by real numbers greater than zero, 3 classes, 80% datasets were taken for the training sample and 20% for the test sample. In this experiment no data standardization was performed.

1. *Kohonen network.* At the beginning of the study, the Kohonen network was trained. The number of output neurons in the network is 3. In this way, after preprocessing by this network, the data dimension decreased by more than 4 times. Network training time is 3.6 ms. On large datasets, obviously, time will be longer, but it is not comparable with dozens of minutes, hours, days of training large networks with complex architectures.

2. *Perceptron*. Reducing the number of input neurons to 3 reduced the number of neurons in the hidden layer from 48, as in [7], to 6, that is, 8 times. The activation function of the neurons of hidden layer is logistic sigmoid and of output layer is softmax function. The optimization algorithm is Adam. Loss function is crossentropy.

3. *Radial-basis function network (RBFN)*. The number of neurons in the hidden layer is 3, decreased by half, compared with [7]. As a radial basis function, a Gaussian function was selected. Gaussian function centers of each neuron of the hidden layer is initialized by centers of 3 clusters, found using the k-means algorithm in the training sample. The optimization algorithm is Adam. Loss function is crossentropy.

4. *Counter propagation network (CPN)*. The number of neurons in the input layer is 3, in the hidden layer is 3. Before the start of training, input vectors were normalized. The weights of the Kohonen layer were initialized with random values from the interval (0, 1) and normalized.

5. *Probabilistic neural network (PNN)*. The number of neurons in the input layer is 3, in the first hidden layer is 142, in the second hidden layer is 3.

6. *NEFClassM*. Number of input neurons is 3. For each feature three initial fuzzy sets were defined with the names “small”, “medium”, “large”. In the rule layer there are 3 neurons. From the trained rule base, one best rule was obtained for each class. Number of output neurons is 3. The maximum number of generated rules is 50 instead of 40 as in [7]. The parameters of fuzzy sets were trained by the gradient method.

7. *Naïve Bayes Classifier*. Distribution functions are normal distributions. Priors is a ratio of the number of samples in the class to the total number of samples.

The learning results of the modules and all the networks described above are presented in Table 1.

TABLE I. NUMBER OF MISCLASSIFIED SAMPLES OF TRAINED MODULES AND INDIVIDUAL NETWORKS

List of NN	Without Kohonen network		With Kohonen network (module)	
	Train	Test	Train	Test
Perceptron	94	28	6	0
RBFN	87	22	5	0
CPN	42	10	20	2
PNN	17	9	7	1
Naïve Bayes	7	2	5	0
NEFClassM	28	7	8	0

As you can see, the simplification of the architecture and the lack of preliminary standardization significantly affected the accuracy of the underlying networks. At the same time, the preprocessing by the Kohonen network gave very good results that exceed the results presented in [7].

Subsequently, individual contributions of each network [7] were found and the ensemble pruning operation [7] was performed. The results obtained are shown in Table 2.

TABLE I. NUMBER OF MISCLASSIFIED SAMPLES WITH PRUNED ENSEMBLE OF INDIVIDUAL NN AND MODULES

Pruned NN ensemble	Pruned modules ensemble			
	Train	Test	Train	Test
Naïve Bayes, NEFClassM, PNN	7	3	3	0

As the results showed, the accuracy of the pruned ensemble of modules is higher than the accuracy of an ensemble of individual networks without preprocessing. It also exceeded the accuracy values of the ensemble from [7]. At the same time, thanks to the simplified architecture of the basic networks, the learning time was significantly lower.

## V. CONCLUSION

The results showed that the use of the neural networks module, the first element of which is the Kohonen network, and the second is the basic network, allows to obtain accuracy indicators that exceed the corresponding indicators of individual networks. At the same time, to achieve a given accuracy, the total training time of the module is much lower than the training of a separate network with a complex architecture. Simplification of the topology of the basic networks in the module allowed to reduce the memory occupied by them.

Due to these advantages, the construction of an ensemble of modules of neural networks is a more efficient and fast solution. As the results of the study showed, the pruned ensemble of the modules presented in this report has accuracy indicators that exceed those of the individual networks from [7], while less memory and training time is required.

## REFERENCES

- [1] M. Wozniak, M. Graña and E. Corchado, “A survey of multiple classifier systems as hybrid systems,” in *Information Fusion*, vol. 16, pp. 3-17, March 2014.
- [2] S. S. Tirumala and A. Narayanan, “Hierarchical data classification using deep neural networks,” in *Neural Information Processing*, Springer International Publishing, 2015, pp. 492-500.
- [3] X-F. Gu and L. Liu and J-P. Li and Y-Y. Huang and J. Lin, “Data classification based on artificial neural networks,” International Conference on Apperceiving Computing and Intelligence Analysis, pp. 223-226, 2008.
- [4] M. Fernandez-Delgado, E. Cernadas, S. Barro, D. Amorim, “Do we need hundreds of classifiers to solve real world classification problems?” in *Journal of Machine Learning Research*, 15, 2014, pp. 3133-3181.
- [5] M. Mitchell, *Genetic Algorithms: An Overview*. Complexity, vol. 1, pp. 31-39, 1995.
- [6] T. Kohonen and T. Honkela, “Kohonen network,” 2007, accessed: March 2012. [Online]. Available: [http://www.scholarpedia.org/article/Kohonen\\_network](http://www.scholarpedia.org/article/Kohonen_network).
- [7] O. I. Chumachenko and K. D. Riazanovskiy, “Structural-parametric synthesis of neural network ensemble based on the estimation of individual contribution,” in *Electronics and Control Systems*, No 59, pp. 66-77, 2019.
- [8] B. Kosko, “Bidirectional associative memories,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 18, pp. 49-60, January/February 1988.