DOI: https://doi.org/10.15276/aait.07.2024.26
UDC 681.3.07:004.8

Reducing the amount of computations at the second stage of an ensemble classifier with stacking on mine-detection drones

Oleg N. Galchonkov [1)]
ORCID: https://orcid.org/0000-0001-5468-7299; o.n.galchenkov@gmail.com. Scopus Author ID: 56081377900
Alexey M. Baranov [1)]
ORCID: https://orcid.org/0009-0002-5951-2636; oleksii.m.baranov@gmail.com
Illia O. Baskov [1)]
ORCID: https://orcid.org/0000-0002-3517-6773, illyabaskov@gmail.com
[1)] Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine

ABSTRACT

Using drones to search for mines is a promising area that allows accelerating the process of demining an area and reducing the danger to people. To increase the probability of detecting mines, drones use sensors with different operating principles. Each type of sensor requires specialized processing, which is carried out at the first stage of the ensemble classifier with stacking. The sensor signals are combined at the second stage of the ensemble classifier, where a multilayer perceptron is usually used as a neural network. Acceleration of terrain survey requires that processing be carried out in real time on the computing equipment of the drone itself. This, in turn, requires a reduction in the amount of computation for all algorithms used on the drone. The article is devoted to reducing the amount of computation when implementing a multilayer perceptron. The original homogeneous structure of the perceptron, when each of the neurons of the previous layer has connections with all neurons of the next layer, is redundant, since it does not take into account the features of the processed data set. The article proposes a method for finding a balance between the dimensionality and number of layers of the perceptron, the time interval between the procedures of thinning connections, the training step and the number of connections removed at a time. Using thinning connections taking into account other parameters allows you to reduce the amount of calculations by 80% or more, while saving and even increasing the quality of classification. There are removed connections that do not make a noticeable contribution to the quality of classification, but introduce additional noise into the perceptron training process and the formation of the output result.

Keywords: Multilayer perceptron; neural network; thinning connections; pruning; weighting coefficients; classification quality

For citation: **Galchonkov O. N., Baranov A. M., Baskov I. O. “Reducing the amount of computations at the second stage of an ensemble classifier with stacking on mine-detection drones”. *Applied Aspects of Information Technology*. 2024; Vol.7 No.4: 370–377. DOI: https://doi.org/10.15276/aait.07.2024.26**

1. INTRODUCTION

One of the most useful auxiliary means used in mine clearance are autonomous unmanned aerial vehicles (quadcopters). They allow you to quickly survey the area without endangering people and indicate the coordinates where the mines are located. To increase the probability of detection, sensors of various physical natures are used. Each of them requires its own specialized processing, the output of which is a vector of signals, characterizing the belonging of an object in the survey zone for solving the problem of recognizing dangerous objects. Machine learning technologies, in particular multi-stage ensemble classifiers, are used to analyze the resulting multidimensional vector. At the first stage, it is proposed to use an ensemble classifier with stacking [1]. Then these vectors go to the second stage of the ensemble classifier, where they are jointly processed by a neural network. This allows you to significantly improve the resulting quality of the classification, compared to the quality of the first stage classifiers [2].

However, since the bulk of the computations in solving the problem of recognizing dangerous objects are performed on the quadcopter equipment itself, this requires meeting strict energy consumption requirements when surveying significant areas of terrain. One way to meet these requirements is to minimize the amount of computations in each node of the processing algorithm, in particular, in the neural network of the second stage of the ensemble classifier. A multilayer perceptron is usually used as such a network [2].

Despite the fact that a multilayer perceptron is a universal approximator [3], the full connectivity between the layers of neurons is redundant. The number of connections, depending on a specific improvement [5]. However, the resulting quality of the trained neural network depends on many practical task, can be significantly reduced without losing the resulting quality [4], or even with some interacting factors: the rate of redundancy reduction (prunning), the speed of training, the total number of remote connections. The nature of the interaction of these factors and the degree of influence on the resulting characteristics of the neural network depend significantly on the specific problem being solved. Therefore, the development of a method for thinning the structure of a multilayer perceptron during training is relevant, which allows for the best resulting characteristics.

2. ANALYSIS OF LITERATURE DATA

At present, a large number of approaches have been developed that can be applied to thinning connections in a multilayer perceptron.

In [6], the Knowledge Distillation approach is proposed. A fully connected neural network is pre-trained, and then special target functions are applied to its weight coefficients, allowing one to determine which connections can be removed. However, in practice, this approach is characterized by increased complexity and the lack of guarantees that important connections will not be removed, which will result in a decrease in the quality of classification [7].

In [8], it is proposed to remove connections corresponding to small eigenvalues of the matrix of weight coefficients (Low-rank Decomposition). However, with a large number of trained weight coefficients, the matrix size is also large. Calculating the eigenvalues and eigenvectors of such matrices is a complex computational task. Errors in calculating the smallest eigenvalues lead to deterioration in the quality of classification after thinning.

Since a large number of deep neural networks are currently used in various applications, structural thinning methods have been developed [9]. Unlike unstructured thinning, where random connections are removed, this approach allows preserving the ordered structure of the neural network. This is necessary for efficient implementation on computing resources. However, for a simple multilayer perceptron architecture, unstructured thinning can also be effectively used, providing higher performance.

A large number of works are devoted to thinning during initialization [10]. During initialization, a mask is imposed on the general architecture of connections in the neural network, which determines the thinning. In fact, this is also a structural thinning. For complex neural networks, this approach is justified, but for a simple architecture of a multilayer perceptron, it does not allow realizing the maximum result from thinning.

As an alternative to thinning connections, in the work [11] to reduce the volume of calculations it is proposed to replace floating point operations with fixed point operations and reduced representation bit depth (Parameter Quantization). The disadvantages of this approach include the fact that it is only effective when implemented on programmable logic matrices, and also the fact that any errors in scaling signals and weight coefficients lead to a significant decrease in the quality of classification.

The use of the matrix of second derivatives of the loss function to identify ineffective links and remove them is proposed in [12]. This approach is also associated with a large amount of computation. In addition, this approach does not take into account the possibility of redistributing the contribution of the remaining links to the classification quality after removing a certain number of links. This makes iterative use of a less complex loss function relevant in the process of training a multilayer perceptron [5].

The specified approaches consider thinning of connections in a multilayer perceptron without taking into account the relationship between thinning and training, the speed of removing connections and the possibility of redistributing the contribution to the resulting quality of classification of the connections remaining after thinning. Therefore, it is relevant to develop a method for reducing the amount of calculations in a multilayer perceptron that takes into account the impact on the quality of classification of such interrelated parameters as the size and number of internal layers of the perceptron, the speed of training, the period of prunning and the number of connections removed at one time.

3. THE PURPOSE AND OBJECTIVES OF THE STUDY

The aim of the work is to develop a method for reducing the amount of computation for a multilayer perceptron while maintaining the quality of classification by thinning out connections during its training, taking into account finding a balance between such parameters as the size and number of internal layers of the perceptron, the training step, the pruning period, and the number of connections removed at a time.

4. MATERIALS AND METHODS OF RESEARCH

4.1. Description of the neural network and ITS training algorithm

For the sake of certainty, when developing the methodology, we will use a multilayer perceptron designed to classify handwritten digits in 28x28 pixel images from the MNIST dataset [13]. This neural network contains an input layer of dimension 784, two hidden layers of dimensions 256 and 128, and an output layer of dimension 10. The sigmoid is used as the activation function in all layers. The training algorithm is stochastic gradient descent with a constant step.

The formation of output signals of the neural network and the learning algorithm are described by the following equations [14]:
input signal of the first hidden layer

$$X_{in-h1} = W_{in-h1} \times X_{in} \;, \qquad (1)$$

where X_{in} is vector of input signals of a neural network, with dimensions $n_{in} = 784$; W_{in-h1} is matrix of weight coefficients between the input layer and the first hidden layer, of dimension $n_{h1} \times n_{in}$; $n_{h1} = 256$ is number of neurons in the first hidden layer, output signal of the first hidden layer

$$X_{o-h1} = f_{act}(X_{in-h1}) \;, \qquad (2)$$

where $f_{act}(x) = 1/(1+e^{-x})$ is activation function, input signal of the second hidden layer

$$X_{in-h2} = W_{h1-h2} \times X_{o-h1} \;, \qquad (3)$$

where W_{h1-h2} is matrix of weight coefficients between the first and second hidden layers, with dimension $n_{h2} \times n_{h1}$; $n_{h2} = 128$ is number of neurons in the second hidden layer, output signal of the second hidden layer

$$X_{o-h2} = f_{act}(X_{in-h2}) \;, \qquad (4)$$

input and output signals of the output layer respectively

$$X_{in-o} = W_{h2-o} \times X_{o-h2} \;, \qquad (5)$$

$$X_{out} = f_{act}(X_{in-o}) \;, \qquad (6)$$

where W_{h2-o} is matrix of weight coefficients between the second hidden layer and the output layer, of dimension $n_o \times n_{h2}$; $n_o = 10$ – number of neurons in the output layer (number of neural network outputs), vector of errors at the output of the neural network (dimension n_o)

$$E_{out} = X_{tar} - X_{out} \;, \qquad (7)$$

where X_{tar} is vector of known correct output signals of a neural network, of dimension n_o, contains all zeros, except for 1 in the place that matches the digit shown in the input image, error vector at the output of the second hidden layer (dimension n_{h2})

$$E_{h2} = W_{h2-o}^{T} \times E_{out} \;, \qquad (8)$$

error vector at the output of the first hidden layer (dimension n_{h1})

$$E_{h1} = W_{h1-h2}^{T} \times E_{h2} \;, \qquad (9)$$

weight update equations

$$W_{h2-o} = W_{h2-o} + \mu((E_{out} \odot X_{out} \odot (1 - X_{out})) \times X_{o-h2}^{T}) \;, \qquad (10)$$

$$W_{h1-h2} = W_{h1-h2} + \mu((E_{h2} \odot X_{o-h2} \odot (1 - X_{o-h2})) \times X_{o-h2}^{T}) \;, \qquad (11)$$

$$W_{in-h1} = W_{in-h1} + \mu((E_{h1} \odot X_{o-h1} \odot (1 - X_{o-h1})) \times X_{in}^{T}) \;, \qquad (12)$$

weight update equations "$\odot$" denotes element-wise multiplication, μ is learning step, scalar value.

Thinning out connections in a neural network is accomplished using auxiliary matrices

$$H_{h2-o} \;, H_{h1-h2} \;, H_{in-h1} \;,$$

with the corresponding dimensions

$$n_o \times n_{h2} \;, n_{h2} \times n_{h1} \;, n_{h1} \times n_{in} \;,$$

$$W_{h2-o} = W_{h2-o} \odot H_{h2-o} \;, \qquad (13)$$

$$W_{h1-h2} = W_{h1-h2} \odot H_{h1-h2} \;, \qquad (14)$$

$$W_{in-h1} = W_{in-h1} \odot H_{in-h1} \;, \qquad (15)$$

where are the matrix elements

$$H_{h2-o} \;, H_{h1-h2} \;, H_{in-h1}$$

have single initial values, when thinning out any connection the corresponding unit is replaced by zero.

Every other *L* eras of learning *k* percent of non-zero matrix elements H_{h2-o} , H_{h1-h2} , H_{in-h1} , corresponding to the elements of the weighting coefficient matrices W_{h2-o} , W_{h1-h2} , W_{in-h1} with the smallest modules, are reset.

One era – training by 60000 images.

4. 2. Description of the MNIST dataset

A feature of using drones to search for mines is that the drone flies at a relatively low altitude, about two meters above the ground. And the sensors can be located on a suspension located at a distance of several tens of centimeters from the ground [15]. Therefore, there is no need for sensors with a high resolution. For example, cameras with a resolution of 28x28, magnetometers with a noise level of about 0.1%, or sensors with neutron fluxes with the same accuracy can be used. Therefore, the method for improving the quality of classification of a multilayer perceptron can be based on the results of processing a data set with a small amount of data in each of the classified elements. The MNIST data set was used as such a data set in this paper [13].

The MNIST dataset contains 60,000 images of handwritten digits for training the neural network and 10,000 for testing. Each image is 28x28 pixels in size and contains information what the object is in the image. Based on this information, the vector used in equation (7) is calculated. To handle color, the image is reformatted into floating point format and normalized. Examples of digits from the MNIST dataset are shown in Fig. 1.

To regularize the learning process, the input images undergo pre-processing:

– image rotation relative to the center by a random angle in the range from -15 to +15 degrees;

– horizontal shift by a random value in the range from -0.05 to +0.05 of the image size by width;

– vertical shift by a random value in the range from -0.05 to +0.05 of the image size by height;

– image expansion/compression relative to the center by a random coefficient in the range from 0.95 to 1.05.

Such pre-processing ensures the uniqueness of the images used in different learning epochs.

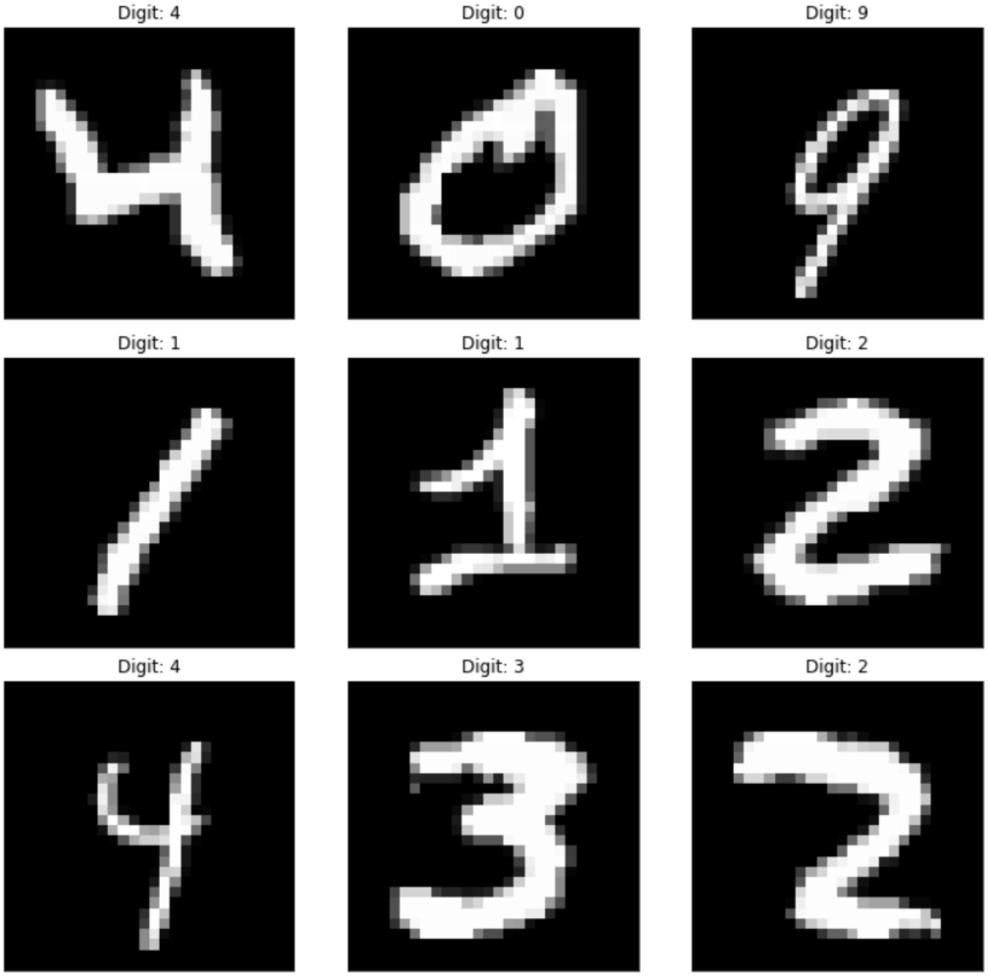

Fig. 1. **Examples of digits from the MNIST dataset**
Source: **compiled by the [16]**

4.3. Experimental data

Extensive experimental studies conducted in [17] revealed the following dependencies of the training and thinning procedure parameters on the resulting characteristics of a multilayer perceptron:

– for a selected neural network configuration and the input data set used, the difference in the resulting characteristics of the neural network when choosing the worst and best set of training procedure parameters can be more than 1 %;

– for a selected neural network configuration and the input data set used, redundancy is completely removed by excluding approximately 80% of the connections, since regardless of the training step and the thinning procedure parameters, further removal of connections leads to worsening of the classification;

– thinning by 1 % after each training epoch provides a slightly better result than thinning by 2% every two epochs and thinning by 3% every three epochs;

– when thinning connections after each epoch, an increase in the percentage of removed connections leads to worsening of the classification quality;

– there is the best value of the training step, which provides the highest value of classification quality; with a small training step, the neural network does not have time to learn, and with a large one, the training process degrades; the corresponding dependences of the maximum achievable classification accuracy depending on the training step are shown in Fig. 2 and this same range of training steps provides the fastest achievement of these maxima – Fig. 3.

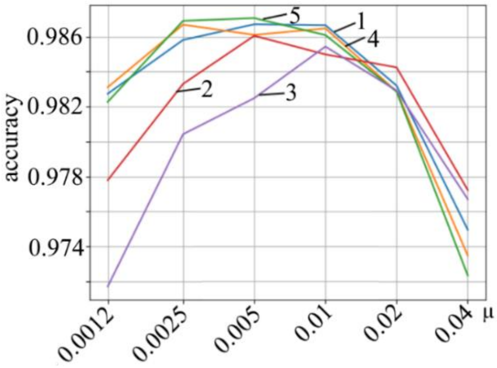

Fig. 2. **Dependence of the maximum classification accuracy on the training step and parameters of the thinning procedure for images from the training set (on the graphs 1 – L=1, k=1; 2 – L=1, k=2; 3 – L=1, k=3; 4 – L=2, k=2; 5 – L=3, k=3)**

***Source:* compiled by the [17]**

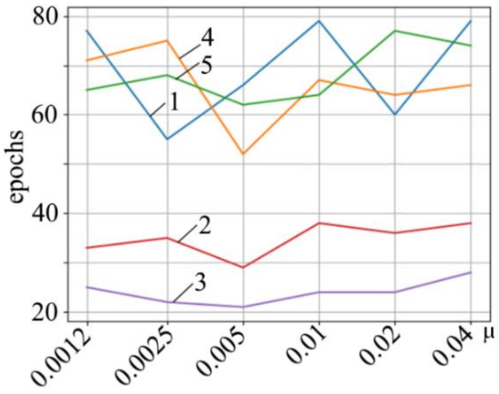

Fig. 3. **Dependence of the number of epochs for achieving maximum classification accuracy on the training step and parameters of the thinning procedure for images from the training set (on the graphs 1 – L=1, k=1; 2 – L=1, k=2; 3 – L=1, k=3; 4 – L=2, k=2; 5 – L=3, k=3)**

***Source:* compiled by the [17]**

5. DEVELOPMENT OF A METHOD FOR IMPROVING THE QUALITY OF CLASSIFICATION OF A MULTILAYER PERCEPTRON BY SETTING PARAMETERS DURING ITS TRAINING

Based on the presence of a clearly expressed extremum in the dependence of the resulting classification quality on the parameters of the training and thinning procedure (Fig. 2), the following steps of the methodology can be formed.

1. For a specific classification problem, based on practical experience and expert assessments, select the number of layers and their dimension for a multilayer perceptron. The training step is selected inversely proportional to the total number of trained weighting coefficients.

2. Refine the value for the training step. To do this, it is necessary to train a multilayer perceptron with the selected value of the training step, smaller and larger. If degradation of the training process is observed (the training curve does not fall smoothly, but is subject to sharp jumps or even an increase), then it is necessary to reduce the training step by an order of magnitude.

If the training curve has a decreasing nature with a steady state at the end, then it is necessary to select the best of the three results and set the corresponding training step. Repeating the training process with a decrease or increase in the training step, it is necessary to achieve the best classification quality.

3. Perform training with simultaneous thinning of weighting coefficients. By changing the training step, the thinning interval, and the number of

simultaneously removed weight coefficients, it is necessary to construct dependencies similar to those shown in Fig. 2. Thinning of weight coefficients must be performed until the classification quality begins to deteriorate.

4. If the best result obtained in step 3 corresponds to the number of removed weight coefficients less than 80 % of the total number of weight coefficients, then it is necessary to increase either the dimensions of the internal layers, or their number, or both. When increasing the number of layers, equations 1-15 are supplemented with similar ones for new layers. After this, it is necessary to repeat steps 2 and 3. By increasing the total number of initial weight coefficients of the multilayer perceptron in this way, it is necessary to achieve a resulting number of removed weight coefficients of 80 % or more.

5. It is necessary to record the best classification quality result in step 4 in the form of matrices of weight coefficients W and auxiliary matrices H in equations 1-15. When using a trained multilayer perceptron, only equations 1-6 are implemented. Since the resulting matrices of weight coefficients consist of 80 % or more zeros, the multiplication of the corresponding vectors X and matrices W is performed as follows. Each i-th element of the resulting vector is found as a scalar product of the original vector X and the vector V containing the corresponding non-zero elements of the i-th row of the matrix W and the corresponding indices for the vector X, indicating which elements of the vector X need to be multiplied by the corresponding elements of the vector V.

6. DISCUSSION OF THE METHODOLOGY

Improving the quality of classification by a multilayer perceptron using the proposed technique is based on the fact that the homogeneous structure of the perceptron does not take into account the features of the processed data set. This is confirmed by the presence of a large number of various neural network architectures that demonstrate significantly higher characteristics than a perceptron with the same number of trainable coefficients.

The technique involves using an increased number of internal layers and their dimensions in the perceptron. As a result of thinning, connections are removed that do not significantly contribute to the quality of classification and are only a source of additional noise in the training and classification process. Since all training and thinning parameters must be balanced to achieve the maximum result, the developed technique is aimed at achieving such a balance.

When obtaining experimental data in [16], the universal MNIST data set was used and no features of this data set are included in the developed technique, training and thinning algorithms. This allows us to hope that the technique will provide an increase in the quality of classification for a wide range of other data sets. It should also be noted that the developed technique is based on the mechanism of thinning out redundant connections in small portions during the training of a multilayer perceptron. When a balance is achieved between the learning rate and the number of connections being removed, the perceptron will have time to retrain the remaining weight coefficients. This should ensure that it is the redundant connections that are removed, and the remaining connections will implement the adjustment to a specific input data set. Therefore, it can be expected that the developed technique will work similarly when using a variable learning step and other activation functions.

7. CONCLUSIONS

In this work, in order to increase the energy efficiency of information analysis when recognizing objects in the survey area of mine-detecting drones by reducing the amount of computations at the second stage of an ensemble classifier with stacking, a method for reducing the amount of computations for a multilayer perceptron while maintaining the quality of classification has been developed, which involves thinning out ineffective connections. This allows the resulting perceptron architecture to be matched to the processed data set and thereby improve the classification quality. The proposed method includes specific steps to achieve a balance between the perceptron parameters, the parameters of the learning algorithm, and the parameters of the thinning procedure. The use of link thinning within the framework of the methodology allows us to reduce the volume of calculations by 80 % or more, while maintaining and even increasing the quality of classification. The method is adapted to the needs of mine-detecting drones, which are distinguished by the processing of low-dimensional data, and can be used to work with any other low-dimensional data sets.

REFERENCES

1. Rokach, L. “Ensemble Learning”. Pattern Classificion Using Ensemble Methods (Second Edition). *World Scientific Publishing Co.* Pte. Ltd. Singapore. 2019.

2. Galchonkov, O., Babych, M., Zasidko, A. & Poberezhnyi, S. “Using a neural network in the second stage of the ensemble classifier to improve the quality of classification of objects in images”. *Eastern-European Journal of Enterprise Technologies*. 2022; 3 (9 (117)): 15–21. DOI: https://doi.org/10.15587/1729-4061.2022.258187.

3. Hornik, K., Stinchcombe, M. & White, H. “Multilayer feedforward networks are universal approximators”. *Neural Networks*. 1989; 2 (5): 359–366. DOI: https://doi.org/10.1016/0893-6080 (89)90020-8.

4. Denil, M., Shakibi, B., Dinh, L., Ranzato, M. A. & Freitas, N. “Predicting parameters in deep learning. 2014. DOI: https://doi.org/10.48550/arXiv.1306.0543.

5. Blalock, D., Gonzalez, Ortiz, J. J., Frankle, J. & Guttag, J. “What is the state of neural network pruning?” *Proceedings of Machine Learning and Systems*. 2020. p. 129–146. DOI: https://doi.org/10.48550/arXiv.2003.03033.

6. Hinton, G., Vinyals, O. & Dean, J. “Distilling the knowledge in a neural network”. *In NIPS Deep Learning and Representation Learning Workshop.* 2015. DOI: https://doi.org/10.48550/arXiv.1503.02531.

7. Wang, Z., Li, F., Shi, G., Xie, X. & Wang, F. “Network pruning using sparse learning and genetic algorithm”. *Neurocomputing*. 2020; 404 (1): 247–256. DOI: https://doi.org/10.1016/j.neucom.2020.03.082.

8. Li, Y., Gu, S., Mayer, C., Gool, L. V. & Timofte, R. “Group Sparsity: The Hinge between filter pruning and decomposition for network compression”. 2020. p. 8015-8024 DOI: https://doi.org/ 10.1109/CVPR42600.2020.00804.

9. He, Y. & Xiao, L. “Structured pruning for deep convolutional neural networks: A survey”. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* 2023; 46 (5): 2900–2919. DOI: https://doi.org/10.1109/TPAMI.2023.3334614.

10. Lee, N., Ajanthan, T. & Torr, P. H. S. “Snip: Single-shot network pruning based on connection sensitivity”. *International Conference on Learning Representations*. 2019. DOI: https://doi.org/10.48550/arXiv.1810.02340.

11. Qiu, J., Wang, J., Yao, S., et al. “Going deeper with embedded FPGA platform for convolutional neural network”. *FPGA’16*. Monterey, USA. 2016. p. 26–35. DOI: http://dx.doi.org/10.1145/ 847263.2847265.

12. LeCun, Y., Denker, J. S. & Solla, S. A. “Optimal brain damage”. *Advances in Neural Information Processing Systems 2.* 1990. p. 598–605. – Available from: http://yann.lecun.com/exdb/publis/pdf/lecun-90b.pdf.

13. LeCun, Y., Cortes, C., Burges, C. J. C. “The MNIST database of handwritten digits”. – Available from: http://yann.lecun.com/exdb/mnist.

14. Galchonkov, O., Nevrev, A., Glava, M. & Babych, M. “Exploring the efficiency of the combined application of connection pruning and source data pre-processing when training a multilayer perceptron”. *Eastern-European Journal of Enterprise Technologies. Information and Controlling System*. 2020; 2 (9 (104)): 6–13. DOI: https://doi.org/10.15587/1729-4061.2020.200819.

15. Florez-Lozanoa, J., Caraffinib, F., Parraa, C. & Gongorab, M.“Cooperative and distributed decision-making in a multi-agent perception system for improvised land mines detection”. *Information Fusion*. 2020; 64: 32–49. DOI: https://doi.org/10.1016/j.inffus.2020.06.009.

16. Galchonkov, O., Nevrev, A., Shevchuk, B. & Baranov, N. “Definition of the influence of the choice of the pruning procedure parameters on the quality of training of a multilayer perceptron”. *Eastern-European Journal of Enterprise Technologies*. 2022; 1 (9 (115)): 75–83. DOI: https://doi.org/10.15587/1729-4061.2022.253103.

Conflicts of Interest: The authors declare that they have no conflict of interest regarding this study, including financial, personal, authorship or other, which could influence the research and its results presented in this article

Financing: The study was conducted with financial support from grant No. 131/0161 of the National Research Foundation of Ukraine

Use of artificial intelligence: The authors used artificial intelligence technologies within acceptable limits to provide their own verified data, which is described in the research methodology section.

Received 10.09.2024
Received after revision 11.11.2024
Accepted 15.11.2024

DOI: https://doi.org/10.15276/aait.07.2024.26
УДК 681.3.07: 004.8

Зменшення об'єму обчислень на другому ступені ансамблевого класифікатора зі стекінгом на дронах-міношукачах

Галчонков Олег Миколайович[1)]
ORCID: https://orcid.org/0000-0001-5468-7299; o.n.galchenkov@gmail.com. Scopus Author ID: 56081377900
Баранов Олексій Миколайович[1)]
ORCID: https://orcid.org/0009-0002-5951-2636; oleksii.m.baranov@gmail.com
Баськов Ілля Олександрович[1)]
ORCID: https://orcid.org/0000-0002-3517-6773; illyabaskov@gmail.com
[1)] Національний університет «Одеська політехніка», проспект Шевченка, 1. Одеса, Україна

АНОТАЦІЯ

Використання дронів для пошуку мін є перспективним напрямом, що дозволяє прискорити процес розмінування місцевості та зменшити небезпеку для людей. Для підвищення ймовірності виявлення мін на дронах використовують різнорідні за принципом дії датчики. Кожен із типів датчиків вимагає спеціалізованої обробки, що здійснюється на першому ступені ансамблевого класифікатора зі стекінгом. Об'єднання сигналів датчиків проводиться на другому ступені ансамблевого класифікатора, де зазвичай як нейронна мережа використовується багатошаровий персептрон. Прискорення обстеження місцевості вимагає, щоб обробка здійснювалася у реальному масштабі часу на обчислювальному обладнанні самого дрону. Це, своєю чергою, вимагає зменшення обсягу обчислень всіх алгоритмів, використовуваних на дроні. Стаття присвячена зменшенню обсягу обчислень під час реалізації багатошарового персептрона. Вихідна однорідна структура персептрона, коли кожен з нейронів попереднього шару має зв'язки з усіма нейронами наступного шару, є надмірною, оскільки не враховує особливості набору даних, що обробляється. У статті запропоновано методику знаходження балансу між розмірністю та кількістю шарів персептрону, інтервалом часу між процедурами проріджування зв'язків, кроком навчання та кількістю зв'язків, що видаляються, за один раз. Використання проріджування зв'язків з урахуванням інших параметрів дозволяє зменшити обсяг обчислень на 80 % і більше, зберігаючи і навіть збільшуючи якість класифікації. Видаляються зв'язки, які не вносять помітного внеску у якість класифікації, але вносять додатковий шум у процес навчання персептрона і формування результату на виході.

Ключові слова: багатошаровий персептрон; нейронна мережа; проріджування; регуляризація; вагові коефіцієнти; якість класифікації

ABOUT THE AUTHORS

Oleg N. Galchonkov - Candidate of Engineering Sciences, Assistent Professor, Information Systems Department. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine
ORCID: http://orcid.org/0000-0001-5468-7299; o.n.galchenkov@gmail.com. Scopus Author ID: 56081377900
Research field: Artificial intelligence; evolution systems; machine learning

Галчонков Олег Миколайович – канд. техніч. наук, доцент кафедри Інформаційних систем. Національний університет ”Одеська політехніка”, пр. Шевченка, 1. Одеса, 65044,Україна

Alexey M. Baranov - Graduate student, Information Systems Department. Odesa Polytechnic National University, 1, Shevchenko Ave. Odesa, 65044, Ukraine
ORCID: https://orcid.org/0009-0002-5951-2636; oleksii.m.baranov@gmail.com
Research field: Artificial intelligence; evolution systems; machine learning

Баранов Олексій Миколайвич - аспірант кафедри Інформаційних систем. Національний університет”Одеська політехніка”, пр. Шевченка, 1. Одеса, 65044,Україна

Illia O. Baskov - Senior Lecturer, Information Systems Department. Odesa Polytechnic National University”, 1, Shevchenko Ave. Odesa, 65044, Ukraine
ORCID: https://orcid.org/0000-0002-3517-6773, illyabaskov@gmail.com
Research field: Artificial intelligence; evolution systems; machine learning

Баськов Ілля Олександрович - старший викладач кафедри Інформаційних систем. Національний університет”Одеська політехніка”, пр. Шевченка, 1. Одеса, 65044,Україна