UDC 004.93
DOI: 10.15587/1729-4061.2018.145586

CONSTRUCTING A METHOD FOR THE CONVERSION OF NUMERICAL DATA IN ORDER TO TRAIN THE DEEP NEURAL NETWORKS

M. Pryshliak
Postgraduate Student*
E-mail: pryshliak.mykhailo@gmail.com
S. Subbotin
Doctor of Technical Sciences, Professor*
E-mail: subbotin.csit@gmail.com
A. Oliinyk
PhD, Associate Professor*
E-mail: olejnikaa@gmail.com
*Department of Software Tools
Zaporizhzhia National Technical University
Zhukovskoho str., 64, Zaporizhzhia,
Ukraine, 69063

Розглянуто відомі види глибоких нейронних мереж, методи їх навчання з учителем, навчання мереж шумозаглушенню, а також методи кодування даних за допомогою зображень. Показано, що глибокі нейронні мережі підходять для ефективного вирішення завдань класифікації, зокрема для медичного і технічного діагностування. Серед глибоких мереж перспективними є згорткові нейронні мережі внаслідок своєї простої структури і застосування спільних ваг, що дозволяє мережі виділяти схожі ознаки в різних частинах зображень. Навчання згорткової мережі може бути недостатньо ефективним для деяких завдань діагностування, тому доцільно розглянути модифікації методу навчання із застосуванням кодування даних і навчанням шумозаглушенню для поліпшення результату.

Запропоновано метод для навчання згорткової нейронної мережі за допомогою числових даних, перетворених в растрові зображення, що підвищує точність мережі при вирішенні задач класифікації і дозволяє застосовувати згорткові нейронні мережі та їх переваги обробки зображень з табличними даними в якості вхідних. Також пропонований метод не потребує внесення додаткових змін в структуру мережі.

Метод складається з чотирьох етапів – нормалізації методом мінімакса, перетворення даних в двовимірні зображення з використанням поплавкового або термометричного методів кодування, генерації додаткових зображень із спотворенням вхідних даних і попереднього навчання глибокої мережі.

Розроблений метод був програмно реалізований та досліджений при вирішенні низки практичних завдань. Результати вирішення практичних завдань технічного та медичного діагностування показали ефективність методу при незначній кількості результуючих класів і екземплярів для навчання. Метод може бути корисний при діагностуванні на ранніх стадіях прояви дефекту, коли обсяг даних для навчання обмежений

Ключові слова: згорткові нейронні мережі, глибоке навчання, перетворення формату, растрові зображення

1. Introduction

The tasks on diagnosing are common in technical and medical applications.

The essence of diagnosing implies solving a classification problem: based on data acquired from a set of sensors as the diagnosing features, it is necessary to relate the diagnosed object to one of the possible classes that characterize an overall state of the diagnosed object or the location and type of a defect in it.

Solving the tasks on technical diagnosing makes it possible to timely detect and prevent failure or malfunctioning of the equipment and the manufactured or operated products. Solving the tasks on medical diagnosing enables diagnosing a patient either without the participation of a physician or to validate the diagnosis made by a physician.

In this case, because of the lack or an insufficient number of experts that can at present diagnose the appropriate objects, it is a relevant task to automate the diagnosing process.

In order to automate decision-making in tasks on technical and medical diagnosing, it is necessary to have a diagnostic model.

To construct diagnostic models, statistical methods are employed (cluster analysis, Bayesian method), decision trees, neural networks (NN), neural-fuzzy networks, and others [1]. Among the most promising methods are NN, specifically deep NN – the layered, non-linear architectures that make it possible to process input at high speed and to obtain the resulting compact models. A deep NN can be trained to solve more difficult problems, however, an increase in the depth of a network leads to the complication of its learning process [2].

There are many varieties of deep NN [3], among which the simplest ones in terms of structure are the convolutional NN (CNN). An advantage of CNN is that they are capable to separate the high-level features in data while sifting out nonessential details. However, the drawback of CNN is the large number of network parameters (which are typically selected empirically) [4]. The task on classification is also resolved by deep belief networks (DBN) [5, 6], although their learning methods are very slow since they come down to the greedy sequential training of all layers in a network. Therefore, in order to solve the diagnosing problems, it is advisable to choose the architecture of CNN.

When solving the tasks on diagnosing, the available values for the diagnostic features in a general case can be assigned as an array of real values. In doing so, to process them in CNN, input data must be represented in the form of images – the two-dimensional arrays of discrete values. Therefore, in order to apply CNN in diagnosing problems, they must be modified. This approach was considered in papers [7–9], although they employed more complex networks than CNN in terms of structure.

Therefore, it is a pressing task to construct new, and improve existing, methods for building diagnostic models based on CNN in order to enable the automated construction of model based on observations and to ensure high accuracy of diagnosing while reducing computational costs.

2. Review of known methods of data conversion and NN learning

The basic principles of CNN operation are described in papers [3, 10]. Studies [7–9] considered methods for encoding numerical data using images and solving a classification problem.

Paper [7] reports results of research into the ensemble NN (known as «associative memory») and shows that their advantage is simplicity and clarity of the operation of the network. However, since the ensemble networks are based on the Hebb hypothesis, methods for training these networks are difficult to implement in the error backpropagation networks.

Study [8] addressed the conversion of sensor data into images with the subsequent transfer training of a deep NN. It is shown that NN, which were trained in advance to solve tasks in one field, can be effectively trained to resolve a target problem from another field. Paper [8] failed to tackle the issues related to the application of data encoding when solving problems with sensory data, especially those that lack the obvious method for converting data into the graphical form.

Paper [9] examined a method to normalize text data for their subsequent conversion into graphical data. It is shown that the application of the min-max normalization makes it possible to retain the scale of data and generates the entire data set in the same range that is critical for certain tasks. Study [9] left unresolved the issue on converting data into color images rather than the gray ones, however the application of color images could increase the overall complexity of the training process. In addition, one should consider encoding the images not only into shades of grey, but into the binary (two-color) images as a simpler encoding method.

So far, the easiest and most popular method of supervised CNN training is an error backpropagation method, showing good results in solving classification problems [10]. This method is easy for parallel computation, which is why it is effective for calculations using the graphical adapters and distributed systems.

To obtain better results, there are methods proposed in papers [11, 12], which come down to adding a resistance to the network against distortions – adding motion blur and defocusing, and adding the Gaussian noise. The application of these approaches for CNN training can improve the network's accuracy of image classification. However, methods reported in [11, 12] contain the unresolved issue: given a small training sample volume, a deep network is capable of overfitting.

A possible variant to overcome the issues related to overfitting is the augmentation – adding additional images, based on the distorted starting ones, to the training dataset. In this case, the distorted images are shuffled in the sample with the starting ones, thereby improving the overall accuracy of the network. Application of augmentation also allows a better network training using small training datasets. This approach was employed in paper [13].

The above suggests that the development of such a CNN training method is expedient that would combine advantages of the considered methods in order to better train a network to solve a classification problem. Specifically, the method must be simple, must encode input data into images for utilizing the learning advantages of CNN and to supplement an input sample with data for stabilization of the training process.

3. The aim and objectives of the study

The aim of this work is to construct a method for training the deep neural networks, designed to work with two-dimensional graphical data, using data in the arbitrary format to train a network.

To accomplish the aim, the following tasks have been set:

– to compare the effectiveness of float and thermometric methods when encoding data for training NN;

– to test experimentally the method proposed for solving the problems on classification – technical and medical diagnosing;

– based on the results of experiments, to determine the efficiency, advantages, and disadvantages of the proposed method.

4. A method of data conversion and training a deep NN

We propose a method consisting of several methods – normalization, conversion of input data, and preliminary training of NN. Within it, each line of the input data in the format of an array of numbers undergoes the normalization first, and then it is converted into an image. Thus, by having a table with input data, we obtain a resulting set of images (one line of data is a single image), which can then be used in order to train deep neural networks.

The proposed method consists of four phases – normalization, data conversion, noise generation, and preliminary training of a network.

At the normalization stage, we determine for each column with feature values the minimum and maximum values by using equations (1) and (2) that will be applied for the normalization.

$$Min_y = \min(v_y^1, v_y^2, ..., v_y^n),\ y = 1...m, \tag{1}$$

$$Max_y = \max(v_y^1, v_y^2, ..., v_y^n),\ y = 1...m, \tag{2}$$

where v_y is the starting value of the feature, y is the ordinal number of the feature, m is the number of features, n is the number of instances. Next, the feature values are normalized by a min-max method so that they are within [0; 1]. The normalized value for the feature is assigned by equation (3):

$$v'_y = \frac{v_y - Min_y}{Max_y - Min_y}, \tag{3}$$

where v'_y is the normalized value for feature y, v_y is the starting value for the feature, Min_y and Max_y are the limit values for each feature calculated in advance.

At the conversion stage, we create N binary (two-color) bitmap images, where N is the number of instances. The following approaches are employed in order to convert numerical data into images.

The first approach is to generate an image the size of $K \times M$ pixels, whose features' values are encoded using a thermometric coding [7], where K is the width of the image, M is the number of features. The value of K defines the number of pixels that are sufficient to encode any feature of the input data. Each row of the image is filled with horizontal lines from the left edge. The length of the line relative to the image width is equal to the normalized value for the feature – expression (3). In the mathematical notation, a probability to fill the pixel with coordinates $\{x; y\}$ in the image is assigned using equation (4):

$$p_{x,y} = \begin{cases} 0, x \le v'_y w \\ 1, x > v'_y w \end{cases}, \; y = 1 \ldots m, \tag{4}$$

where $p_{x,y}$ is the value for a pixel in the image, x and y are the coordinates of this pixel, v'_y is the normalized value for feature y, w is the width of the image, m is the number of features.

Another approach is the generation of an image the size of $K \times M$ pixels, whose features' values are encoded using a float coding [7]. The image is formed by a method similar to the thermometric encoding. The difference is that, instead of a horizontal line, a value for each feature is represented by a group of pixels separating the line to a ratio assigned by the normalized input value for feature v'_y. In the mathematical notation, a condition for constructing an image is assigned by the following equation (5):

$$p_{x,y} = \begin{cases} 0, x - k \le v'_y w \le x + k, \\ 1, x - k > v'_y w, \\ 1, x + k < v'_y w, \end{cases} \quad y = 1 \ldots m, \tag{5}$$

where $p_{x,y}$ is the value for a pixel in the image, x and y are the coordinates of this pixel, v'_y is the normalized value for feature y, w is the width of the image, m is the number of features, k is the size of a float.

At the noise generation stage, additional images are generated based on the normalized values for features. An image generation is performed applying the methods described earlier, only the value for each feature v'_y is shifted at some random value.

At the stage of pre-training, the learning rate is given by a value that is twice the learning rate, used in the subsequent adjustment of the network using the starting data. This provides for a significant gain in an initial increase in the probability that a network makes a correct decision. The subsequent training (fine-tuning) of the network employs the error backpropagation.

Given the proposed method, it is possible to create its simplified versions: without generating distorted images or without pre-training (only the data conversion into an image is performed and a network is trained applying an error backpropagation method).

In order to investigate the properties of the considered methods, we conducted experiments aimed to solve practical classification problems. Computational experiments were performed in the Matlab R2017b programming environment utilizing its capabilities to create CNN with an arbitrary structure.

The source data were composed of several datasets obtained from public repositories. The first set is the data on engines diagnosing [14], consisting of 58,509 instances, each instance contains 48 features. Elements must be related to one of 11 classes, depending on the malfunction. The second set is for diagnosing breast cancer [15]. This set consists of 32 features and 569 instances. The result of solving the problem is a diagnosis – the presence or absence of cancer. The third set is to diagnose ultrasonic meters for liquids [16], consisting of 173 features and 540 instances in total. In the paper [17] that employed this set of data did not consider two meters (marked «C» and «D» in the original problem), so they will be considered in the further experiments. In this case, the number of resulting classes was reduced to two – the presence or absence of a defect.

During experiments, the following CNN architecture was used for all problems: an input layer of the network, a convolution layer with 8 filters the size of 8×8, a layer of nodes of linear rectification, a subsampling layer with blocks the size of 2×2, a fully connected layer, a SoftMax layer and layer of classification. For each task, this architecture differed by the size of the input and the fully connected layers. For the first problem, the input layer size is 48×30, fully-connected – 11, for the second – the input layer size is 30×30 and 2 neurons in the fully-connected one, for the third – input layer size is 43×30 and the fully-connected one – 2 neurons.

The first experiment implied comparing two methods of encoding numerical data – a float and a thermometric [7] encoding methods, in order to solve the first problem – engines diagnosing.

The second experiment was conducted using three different learning methods. The first method is the pre-training using distorted data with subsequent tuning. Within it, the source data were supplemented by the same amount of input data, but with the addition of noise (that is, the training sample increases two-fold). The pre-training proceeded at a double learning rate, followed by the network adjustment – training at normal rate. The second method also employed pre-training at double rate, but excluding the data supplemented with noise. The third method trained the network at normal rate throughout the entire experiment.

The choice of training parameters implied the following. The first two methods employed pre-training at a rate of 0.02 and the adjustment at a rate of 0.01; in the third method, the rate was constant and was equal to 0.01. The maximum number of pre-training epochs in the first method was two times less than that in the second (because the dataset for pre-training with distorted images was twice larger than the original). The maximum number of epochs for the adjustment was the same for the first two methods, for the third it was taken as the sum of the number of epochs of pre-training and adjustment in the second method. Training was terminated in advance if we did not observe any decrease in the validation error for 30 consecutive iterations in the test dataset.

The input data for all models were divided into two samples – training (60 %) and test (40 %). Data were converted using the methods proposed in the format of black and white images. In the process of training, we selected the best epoch (the first epoch that showed the best validation result on the test sample). At the end of each epoch, at the stage of training, we measured time and computed a probability estimate of making a correct decision for both samples (training and test).

We describe, as an example, solving the task on technical diagnosing the ultrasonic meters. At the first stage, the input data in a tabular format are converted, by using one of the two considered encoding methods (float or thermometric), into two-dimensional images, an example of which is shown in Fig. 1. In the image, each feature consists of a horizontal bar in the case of the thermometric coding (Fig. 1, *a*) or a small segment, shifted horizontally, in the case of the float method (Fig. 1, *b*).

a *b*

Fig. 1. Examples of images acquired from the encoding of the same instances of numerical data using different methods: *a* – thermometric method of encoding, *b* – float method

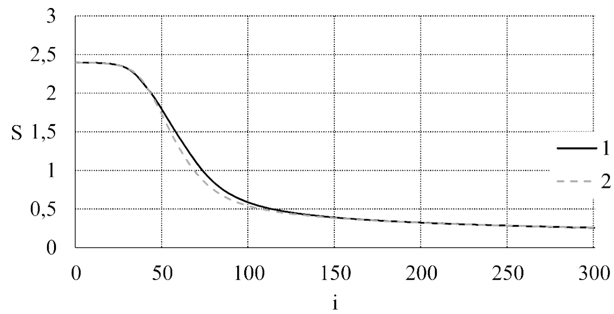

Fig. 3. Dependence chart of validation error *S* on training iteration number *i*: 1 – float method of encoding 2 – thermometric method

At the next stage, the resulting set of images is supplemented with new images that have slightly skewed values. For images the size of 30×30 pixels such distortions are almost invisible to the eye and represent the offset of features' values by 1–2 pixels along the feature value axis (horizontally).

The next stage is the pre-training of CNN over several iterations with a two-fold increase in the learning rate, using both the original data and the distorted ones.

Table 1

Numerical results of experiments over the methods for data encoding

Encoding method	Number of training epochs	Training time to the limit epoch, s	Number of the best epoch	Training time to the best epoch, s	Estimate of the probability for making a correct decision, %
Float	700	2,025.64	685	1,983.10	97.96
Thermometric	700	2,036.69	700	2,036.69	96.73

The learning rate then reverts to its original value and the network is trained on the original data until it reaches the limit of iterations or an estimate of the probability of making a correct decision begins to subside.

The result of the implementation of all of the stages is the formed neural-network model that is capable to categorize the image sent to its input according to one of several output classes, which is the solution to a diagnosing task.

Experiments were performed employing the following hardware: Intel Core i5-4570K processor, 32 GB RAM DDR3-1600, graphic adapter GeForce GTX 1060 6GB (used in all experiments).

5. Results of training the models to solve classification tasks

Based on the results of the experiments that we conducted to compare the encoding methods for numerical data, we have constructed the charts for a change in the probability of making a correct decision and validation errors (Fig. 2, 3). Numerical results from the experiments are given in Table 1.

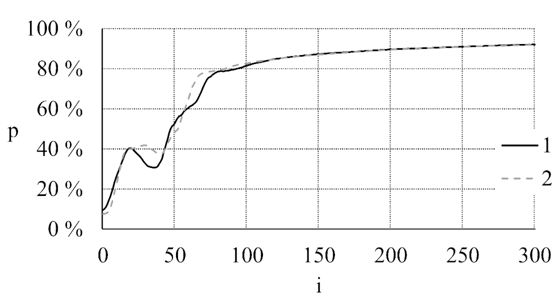

Fig. 2. Dependence chart of estimate of the probability for making a correct decision *p* on the test sample on the training iteration number *i* : 1 – float method of encoding 2 – thermometric method

The result of experiments on CNN training methods is the constructed charts of estimate of the probability of making a correct decision on the test sample for each of the three diagnosing tasks (Fig. 4–6).

For the first task (engine diagnosing), features' values were acquired by measuring the current of working and defective motors [14].

In the second task, we employed data on breast cancer, obtained by converting the results of a biopsy of breast mass into an array of features [15].

The third task applied a set of diagnostic parameters acquired from sensors at several ultrasonic liquid meters [16]. Constraints for experiments are given in Table 2; the results are in Table 3.

Dependence chart of estimate of the probability for making a correct decision on the test sample (Fig. 2) shows that both methods for encoding numerical data into the format of images operate with approximately the same accuracy. The chart of change in the validation error (Fig. 3) demonstrates that for both methods an error in training decreases approximately at the same speed.

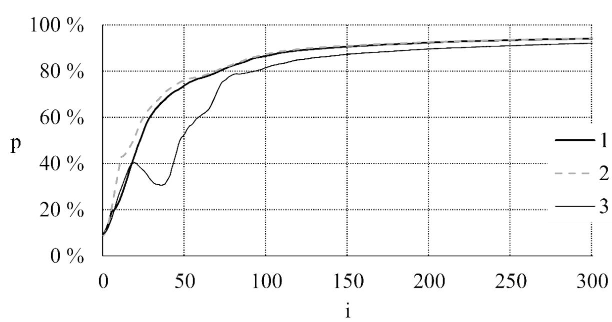

Fig. 4. Dependence chart of estimate for the probability of making a correct decision *p* on the test sample for the first task on the training iteration number *i*: 1 – pre-training with added noise, 2 – pre-training without noise, 3 – without pre-training

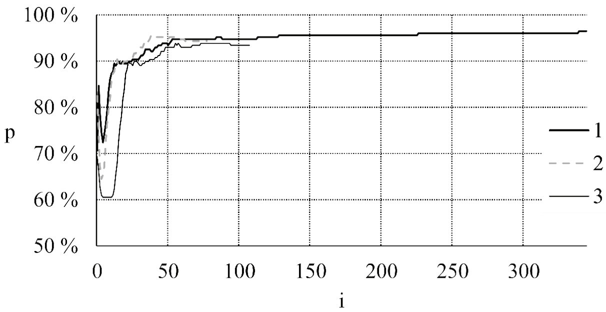

Fig. 5. Dependence chart of estimate of the probability for making a correct decision p on the test sample for the second task on the training iteration number i: 1 – pre-training with added noise, 2 – pre-training without noise, 3 – without pre-training

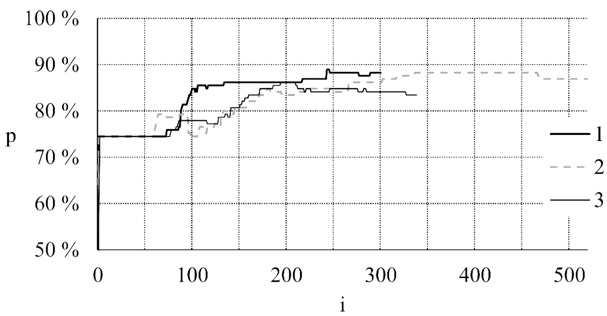

Fig. 6. Dependence chart of estimate of the probability for making a correct decision p on the test sample for the third task on the training iteration number i: 1 – pre-training with added noise, 2 – pre-training without noise, 3 – without pre-training

Charts for the second experiment (Fig. 4–6) show that the preliminary training of a network using additional noise and at an elevated learning rate produces better results than when training at a constant speed. In the charts for the first two tasks, the application of pre-training contributed to the early increase in the likelihood of making a correct decision. While training at a constant speed, the network demonstrated a certain delay in learning.

The estimation chart for the probability of making a correct decision in the first task (Fig. 4) shows that adding an additional noise to the input data does not yield the desired effect.

Table 2

Constraints on training processes in experiments

Training method	Task	Number of pre-training epochs	Number of training epochs
Pre-training with added noise	1	2	700
	2	5	219
	3	10	291
Pre-training without noise	1	4	700
	2	20	19
	3	20	500
Without pre-training	1	–	700
	2	–	54
	3	–	339

Table 3

Results of experiments on practical tasks

Training method	Task	Training time to the limit epoch, s	Number of the best epoch	Training time to the best epoch	Estimate of probability of making a correct decision, %
Pre-training with added noise	1	2,011.52	689	1,974.12	98.03
	2	13.22	163	9.73	96.49
	3	9.17	244	7.45	88.97
Pre-training without noise	1	2,054.93	684	2,009.16	98.02
	2	2.21	20	1.1	95.61
	3	15.61	337	10.12	88.28
Without pre-training	1	2,025.64	685	1,983.10	97.96
	2	3.04	28	1.58	93.86
	3	10.33	195	5.87	86.21

6. Discussion of results of studying the proposed method

Based on the results from the first experiment, it was found that when using a float encoding method, the network learns a classification task slightly better, the probability of making a correct decision on the best epoch for the float method has proved to be higher than that for the thermometric one, as well the learning rate. The thermometric method initially showed higher values for the estimate of probability of making a correct decision, but it could not produce by the end of training any better results than the float method.

Table 1 shows that when applying a float encoding the network learns slightly faster (an increase to 2.6 % of the total training time to the epoch with the best value) and with a higher accuracy (1–1.5 % better). That is why in the second experiment (when solving a classification task) we considered only the float encoding method.

In the task on diagnosing the engines (Fig. 4), the chart shows that the resulting accuracy of the two methods with the pre-training turned out to be almost identical. Given this, we can conclude that for tasks with large training samples or with a large number of resulting classes the pre-training with added noise is inefficient. In this case, it would suffice to train a network in advance at a higher learning rate. As a result, all three methods trained the network with approximately the same accuracy.

Fig. 5 shows results of solving the problem from a public repository on diagnosing breast cancer [15]. Values for the features in a given problem describe the characteristics of cell nuclei and are calculated based on the digitized images acquired as a result of a biopsy of the breast mass. The proposed network training method has also turned out to be more efficient than the simplified methods. When using it, the training was not terminated in advance due to an increase in the validation error (which happened to the other two methods), and, as a result, the model was able to achieve high classification accuracy (higher by 0.9–2.6 %), but used 6–9 times more time.

The chart of estimate of the probability of making a correct decision for the task on diagnosing the ultrasonic meters for liquids (Fig. 6) shows that the proposed method could achieve higher accuracy in solving the problem at earlier iterations. In this case, we obtained a better result for the accuracy of classification compared to the simplified methods. In the best epoch, the accuracy of the network, trained with the pre-training and added distortion, was higher by 0.7–2.7 % than the simple methods. Despite the fact that the relative accuracy and learning rate proved to be higher than for other simplified methods, the network did not demonstrate any outstanding results compared to the other considered tasks.

Tables 2, 3 show that the proposed method improves the accuracy of the resulting model when the datasets are small (up to a thousand of elements), or if the number of the resulting classes is small. In the case of the first task, its application is not effective, although the approach to the conversion of numerical data into images proved to be quite good for this task.

Based on the results from experiments involving the proposed and existing methods of machine learning [18–20], using identical data sets, we compiled Tables 4–5.

Table 4

Comparison of results from existing methods of classification and the proposed method when solving a problem on engine diagnosing

Classification method	Probability of making a correct decision
Multi-dimensional hierarchical networks [18]	86–96 %
DBN with pre-training [19]	94.52 %
DBN with pre-training, self-training, and regularization [19]	96.52 %
Proposed method	98.03 %

Table 5

Comparison of results from existing methods of classification and the proposed method when solving a problem on diagnosing breast cancer

Classification method	Probability of making a correct decision	Training time, s
GRU-SVM [20]	93.75 %	174
Method of support vectors [20]	96.09 %	14
Proposed method	96.49 %	9.73
Softmax-regression [20]	97.65 %	25
Multilayered perceptron [20]	99.03 %	28

Table 4 shows that the proposed method turned out to be 1.5 % more accurate than the previously known machine learning methods when solving classification tasks. Table 5 shows that the proposed method outperforms classical methods in terms of model training time and, as a result, demonstrates good accuracy. However, it does show worse results for accuracy than the method of building classifying models based on a multilayer perceptron. As a consequence, one can conclude that the proposed method is advisable for use to solve the tasks for which learning rate is important, and for which it is expedient to compromise the accuracy of the resulting model.

All this allows us to argue on the possibility of using the proposed method to construct diagnostic classification models based on observation applying CNN. In this case, a classic method of NN training was supplemented with a stage of data conversion that makes it possible to use the convolutional networks in order to process numerical data, while the pre-training and training to suppress noise make a network resistant to distortion. The method has been useful for training a deep NN under conditions of insufficient data for training, without overfitting a network.

Experiments have shown a possibility to apply the proposed method when constructing classifying models using real data sets to diagnose defects in equipment and to categorize diseases, balanced for accuracy and learning rate. That would make it possible to build more complex diagnosing models, spending less time on their training.

Prospects for the further research in this direction imply the application of networks' architectures other than convolutional, as well as the introduction of additional constraints to the training process. In addition, attention should be paid to other methods of image encoding, not covered in this paper, and to the application of different approaches to distortion (Gaussian blur, offset of parts of an image, etc.) when generating images with noise.

7. Conclusions

1. We have proposed a method for training CNN using numerical data that enables the construction of neural models in order to solve a classification problem with sufficient accuracy and learning rate through encoding, added data, and training a network to suppress noise. Data encoding using images has made it possible to apply CNN and their features of visual data analysis for data that do not have any obvious conversion method into a graphical form. Training a network to suppress noise helped stabilize the training process and to train the resulting model to neglect distortions in real data. Supplementation of sample data enables the application of the method at a small number of training instances. By using the proposed method, we performed experiments on methods to encode images, resulting in that we selected, for the further experiments, the float method that demonstrated better accuracy than the thermometric one.

2. The experiments on training the convolutional neural networks to solve three different diagnosing problems have showed that the method is capable to train the models to solve a classification task with an accuracy larger than 85 % (depending on the problem, input data, and conditions to terminate the training process).

3. Based on results from the experiments, it was established that the conversion of numerical data into images makes it possible to construct diagnostic model based on deep NN, designed to work with images without making any additional modifications to their architecture. Additional stages in training have a positive effect on the resulting model, but a marked increase in the accuracy of classification was observed only for models with a small number of classes and instances. Application of data with noise helps a network go beyond a local minimum during training, as well as train the model to ignore random distortions in data.

References

1. Bishop C. M. Pattern Recognition and Machine Learning. New York, 2006. 749 p.
2. Kukačka M. Overview of Deep Neural Networks // WDS 2012: proceedings of 21st Annual Conference of Doctoral Students. Prague, 2012. P. 100–105.
3. Goodfellow I., Bengio Y., Courville A. Deep learning: adaptive computation and machine learning. London, 2016. 775 p.
4. Strigl D., Kofler K., Podlipnig S. Performance and Scalability of GPU-Based Convolutional Neural Networks // 2010 18th Euromicro Conference on Parallel, Distributed and Network-based Processing. 2010. doi: https://doi.org/10.1109/pdp.2010.43
5. Zhou S., Chen Q., Wang X. Discriminative Deep Belief Networks for image classification // 2010 IEEE International Conference on Image Processing. 2010. doi: https://doi.org/10.1109/icip.2010.5649922
6. Liu Y., Zhou S., Chen Q. Discriminative deep belief networks for visual data classification // Pattern Recognition. 2011. Vol. 44, Issue 10-11. P. 2287–2296. doi: https://doi.org/10.1016/j.patcog.2010.12.012
7. Gol'cev A. D. Neyronnye seti s ansamblevoy organizaciey: monografiya. Kyiv: Naukova dumka, 2005. 200 p.
8. Transforming sensor data to the image domain for deep learning – An application to footstep detection / Singh M. S., Pondenkandath V., Zhou B., Lukowicz P., Liwickit M. // 2017 International Joint Conference on Neural Networks (IJCNN). 2017. doi: https://doi.org/10.1109/ijcnn.2017.7966182
9. Sane P., Agrawal R. Pixel normalization from numeric data as input to neural networks: For machine learning and image processing // 2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET). 2017. doi: https://doi.org/10.1109/wispnet.2017.8300154
10. Sozykin A. V. An Overview of Methods for Deep Learning in Neural Networks // Bulletin of the South Ural State University. Series «Computational Mathematics and Software Engineering». 2017. Vol. 6, Issue 3. P. 28–59. doi: https://doi.org/10.14529/cmse170303
11. Zhou Y., Song S., Cheung N.-M. On classification of distorted images with deep convolutional neural networks // 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2017. doi: https://doi.org/10.1109/icassp.2017.7952349
12. Improving the Robustness of Deep Neural Networks via Stability Training / Zheng S., Song Y., Leung T., Goodfellow I. // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016. doi: https://doi.org/10.1109/cvpr.2016.485
13. Salamon J., Bello J. P. Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification // IEEE Signal Processing Letters. 2017. Vol. 24, Issue 3. P. 279–283. doi: https://doi.org/10.1109/lsp.2017.2657381
14. Dataset for Sensorless Drive Diagnosis Data Set. URL: https://archive.ics.uci.edu/ml/datasets/Dataset+for+Sensorless+Drive+Diagnosis
15. Breast Cancer Wisconsin (Diagnostic) Data Set. URL: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)
16. Ultrasonic flowmeter diagnostics Data Set. URL: https://archive.ics.uci.edu/ml/datasets/Ultrasonic+flowmeter+diagnostics
17. Linear dimensionality reduction for classification via a sequential Bayes error minimisation with an application to flow meter diagnostics / Gyamfi K. S., Brusey J., Hunt A., Gaura E. // Expert Systems with Applications. 2018. Vol. 91. P. 252–262. doi: https://doi.org/10.1016/j.eswa.2017.09.010
18. Li L., Dai G., Zhang Y. A Membership-based Multi-dimension Hierarchical Deep Neural Network Approach for Fault Diagnosis // Proceedings of the 29th International Conference on Software Engineering and Knowledge Engineering. 2017. doi: https://doi.org/10.18293/seke2017-074
19. Lee H.-W., Kim N., Lee J.-H. Deep Neural Network Self-training Based on Unsupervised Learning and Dropout // The International Journal of Fuzzy Logic and Intelligent Systems. 2017. Vol. 17, Issue 1. P. 1–9. doi: https://doi.org/10.5391/ijfis.2017.17.1.1
20. Agarap A. F. M. On breast cancer detection // Proceedings of the 2nd International Conference on Machine Learning and Soft Computing – ICMLSC '18. 2018. doi: https://doi.org/10.1145/3184066.3184080