DE GRUYTER OPEN
1862 RIGA TECHNICAL UNIVERSITY
Information Technology and Management Science
ISSN 2255-9094 (online)
ISSN 2255-9086 (print)
December 2017, vol. 20, pp. 20–24
doi: 10.1515/itms-2017-0003
https://www.degruyter.com/view/j/itms

Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets

Pavlo M. Radiuk
Khmelnitsky National University, Ukraine

***Abstract* – A problem of improving the performance of convolutional neural networks is considered. A parameter of the training set is investigated. The parameter is the batch size. The goal is to find an impact of training set batch size on the performance. To get consistent results, diverse datasets are used. They are MNIST and CIFAR-10. Simplicity of the MNIST dataset stands against complexity of the CIFAR-10 dataset, although the simpler dataset has 10 classes as well as the more complicated one. To achieve acceptable testing results, various convolutional neural network architectures are selected for the MNIST and CIFAR-10 datasets, with two and five convolutional layers, respectively. The assumption about the dependence of the recognition accuracy on the batch size value is confirmed: the larger the batch size value, the higher the recognition accuracy. Another assumption about the impact of the type of the batch size value on the CNN performance is not confirmed.**

***Keywords* – Batch size, convolutional neural network, dataset, testing accuracy.**

I. Introduction

In machine learning, a convolutional neural network (CNN) is a class of multilayer artificial neural networks that have successfully been applied to analysing visual images. CNNs are widely used in image and video recognition, recommender systems and natural language processing. The problem of applying these networks is one of the supervised learning tasks [1], and mathematically can be formalized as follows:

$$F\left(\boldsymbol{w}\right)=\min_{\boldsymbol{w}\in\mathbb{R}^n}\left\{\frac{1}{\left|X\right|}\sum_{x\in X} f\left(x,\boldsymbol{w}\right)\right\}, \tag{1}$$

where $F(\boldsymbol{w})$ is a loss function, $\boldsymbol{w}$ is the vector of weights being optimised, n is the dimension of the weight vector $\boldsymbol{w}$, X is a labelled training set, and $f\left(x,\boldsymbol{w}\right)$ is the loss computed from samples $x\in X$ and their labels y. The process of optimising function $F(\boldsymbol{w})$ is also called training of the network. Stochastic Gradient Descent (SGD) and its variants are often used for training convolutional networks. These methods minimise the objective function F by iteratively taking steps of the form:

$$\boldsymbol{w}_{t+1}=\boldsymbol{w}_t-\eta\left(\frac{1}{\left|B\right|}\sum_{x\in B}\nabla f\left(x,\boldsymbol{w}_t\right)\right), \tag{2}$$

where B is a batch sampled from X and $\left|B\right|$ is the batch size, η is the learning rate and t is the iteration index [2]. These methods can be interpreted as gradient descent using noisy gradients, which are often referred to as mini-batch gradients with the specified batch size. SGD and its variants are employed in a small-batch regime, where $\left|B\right|\in X$ and typically $\left|B\right|\in\{16,32,\ldots,512\}$ [3].

The process of training CNN has a great deal of parameters to be set up and adjusted, where the batch size is the most influential one [4]. This parameter represents a number of training samples that will be used during the training in order to make one update to the network parameters. Specifically, the batch size is used when fitting the model, and it controls how many predictions must be made at a time. Summing up the above-mentioned considerations, the batch size impacts the CNN training both in terms of the time to converge and the amount of overfitting, i.e., smaller batch size yields faster computation (with appropriate implementations), but requires visiting more examples in order to reach the same error, since there are less updates per training iteration.

II. Analysis of Related Research

There has been a recent survey in optimisation methods for machine learning, both in the batch and stochastic paradigms. Algorithms like Stochastic Variance Reduced Gradient (SVRG) method [5] and related approaches [6] mix SGD-like steps with some batch computations to control the stochastic noise. Others have proposed to parallelize stochastic training through large mini-batches [7]. However, in such works algorithms of gradient descent are investigated with selected values of hyperparameters in advance.

Research [8] shows advantage of online training to batch training. Online training is the same as batch size equals 1, and batch training is the same as batch size equals to a number of all training datasets. Minibatch training is somewhere between these two approaches and is determined by the batch size. As an acceptation, papers [6] and [9] suggest this number to be set no more than 64. Meanwhile, studies empirically showed in related works [1] and [7] that on the large datasets large batch sizes caused optimisation difficulties, but the trained networks demonstrated good generalisation.

In general, practitioners agree that the optimal value of the batch size parameter for CNN is located in the range of 64 to 512 [10]–[12]. The value is usually set to a power of 2, which is explained by the effective work of optimised matrix operation libraries [9]. However, some papers such as [8], [13], [14] suggest setting batch size equal to multiple of 10, and receiving high values of recognition accuracy on different datasets. Therefore, the question of choosing the optimal batch size weight for diverse datasets remains open and requires further research.

III. Aim and Tasks of the Research

The aim of this paper is to figure out the best batch size for the training speed and the recognition accuracy of CNN. To achieve the aim, the following tasks must be accomplished:

1. To select diverse datasets that will be used to train the neural network.
2. To determine the sequence of batch size values that will be used to train the neural network.
3. To investigate whether the type of the batch size value affects the accuracy of image recognition.
4. To extract the best scores of training accuracy and suggest a factor or a group of factors unifying those scores, related to the batch size values.

IV. Benchmark Datasets

To estimate network training performance, the benchmark classification image problem is conducted on the MNIST and CIFAR-10 datasets. These datasets are widely used to evaluate different CNN architectures [15]. They are easy to use and show satisfactory results for benchmarking.

The MNIST (Mixed National Institute of Standards and Technology) database is used extensively for training and testing machine learning models [16]. The database consists of the pairs, which are "handwritten digit image" and "label". Digit ranges from "0" to "9", meaning 10 patterns in total. Handwritten digit images are grey scale images with pixel size of 28 × 28, labels – actual digit numbers this handwritten digit image represents, it is either "0" to "9" (Fig. 1).

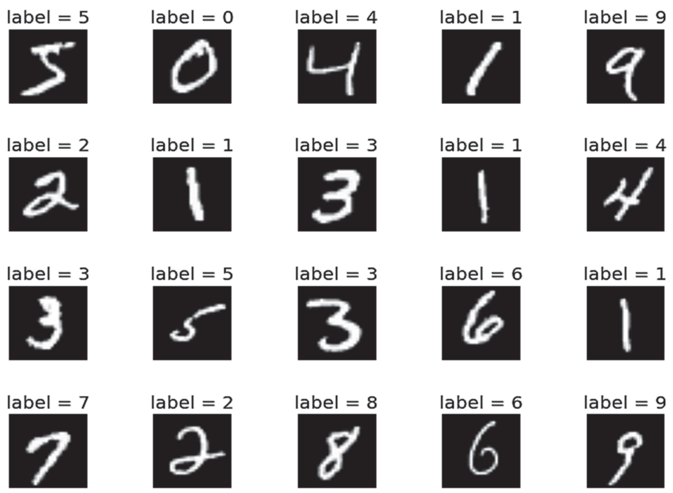

Fig. 1. The MNIST dataset has a training set of 60 000 examples, and a test set of 10 000 examples.Several samples of "handwritten digit image" and its "label" from the MNIST dataset.

Another dataset for training is the CIFAR-10 database, which consists of 60 000 32 × 32 colour images in 10 classes, with 6000 images per class [17]. There are 50 000 training images and 10 000 test images. The dataset is divided into five training batches and one test batch, each with 10 000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class (Fig. 2).

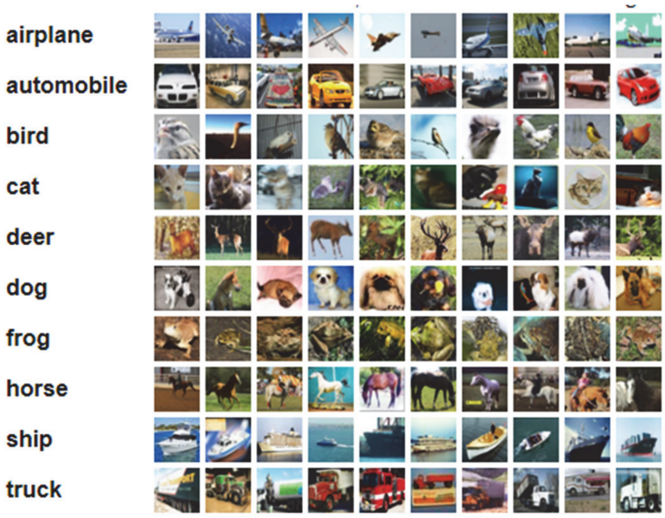

Fig. 2. The variety of colour images from the CIFAR-10 dataset containing 10 image categories (labelled as "airplane", "automobile", "bird", "cat", "deer", "dog", "frog", "horse", "ship", "truck").

In theory, batch size should impact training time and not so much test performance. It can be optimised separately of the other parameters. This is executed by comparing training curves (training and testing error versus amount of training time), after the other parameters and hyperparameters have been selected.

V. Model Architectures

According to the related works, two sequences of the values of the batch size are chosen, namely, numbers to the power of two and numbers multiple of ten. On the whole, 12 batch size values were selected

$$B_k \in \{16, 32, 64, 128, 256, 512, 1024\},$$

and

$$B_l \in \{50, 100, 150, 200, 250\}.$$

To achieve the aim, different CNN architectures were applied to each dataset. This approach would enable to obtain acceptable accuracy of recognition at low cost of time.

The primary challenge was to investigate the impact of batch size on the MNIST dataset. Therefore, a well-known architecture of CNN, called LeNet, was used [18]. It consists of two convolutional layers (ConvLs), two maximum pooling layers (MPLs), two rectified linear unit layers (ReLUs), two fully connected layers (FCLs), and a softmax layer (SML). The spoken architecture is the following:

$$\begin{aligned} &\text{Input} \to \{\text{ConvL}_i \to \text{ReLU}_i \to \text{MPL}_i\}_{i=1}^{2} \\ &\to \{\text{FCL}_j\}_{j=1}^{2} \to \text{SML}. \end{aligned} \tag{3}$$

In order to conduct testing on the CIFAR-10 dataset, a neural network with five convolutional layers was used. To the above mentioned layers, normalisation layers (NLs) were added. This layer normalised the activations of the previous layer at each batch, i.e., applied a transformation that maintained the mean activation close to 0, and the activation standard deviation close to 1. The CIFAR-10 model architecture is presented below:

$$\begin{aligned}&\text{Input} \to \{\text{ConvL}_k \to \text{ReLU}_k \to \text{NL}_k \to \text{MPL}_k\}_{k=1}^{2}\\&\to \{\text{ConvL}_l \to \text{ReLU}_l\}_{l=3}^{5} \to \text{NL}_3 \to \text{MPL}_3\\&\to \{\text{FCL}_m\}_{m=1}^{2} \to \text{SML}.\end{aligned} \tag{4}$$

The selected models were applied using the machine learning framework TensorFlow v. 1.3.0 [19]. The results of the training are displayed with its visualisation toolbox TensorBoard.

VI. Experimental Results and Comparison

The models (3) and (4) are trained using SGD with learning rate of 0.001 and 0.0001 for the MNIST and CIFAR-10 datasets, respectively. In order to effectively and rapidly train the models, performance was evaluated as an average over 5k and 10k iterations for the MNIST and CIFAR-10 datasets, respectively. The networks were executed one time in each experiment. The results of training the network are summarised in Table I.

Figures 3–4 visualise the testing accuracy results. We can see that curves, which describe testing accuracy results, are noisy on the MNIST dataset and smooth on the CIFAR-10 dataset. The curves vary from the smallest batch size value, which is 16, to the largest one, which is 1024.

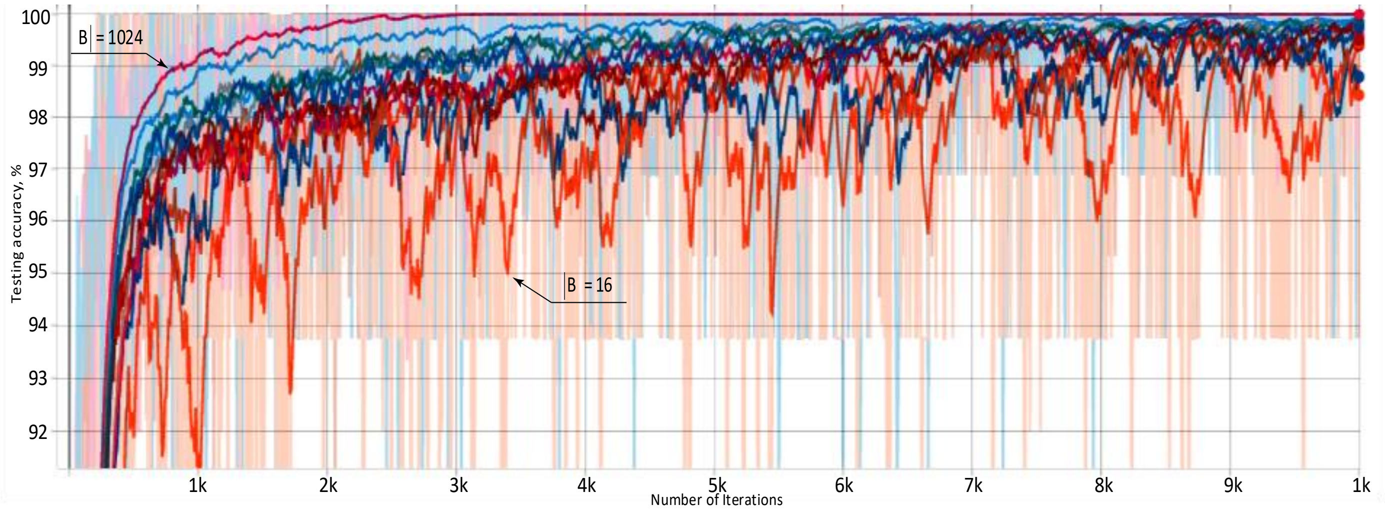

Fig. 3. The testing accuracy of the trained CNN with sequences B_k and B_l on the MNIST dataset. The larger the batch size value, the more smooth the curve. The lowest and noisiest curve corresponds to the batch size of 16 examples, the highest and the smothest one – to the batch size of 1024 examples.

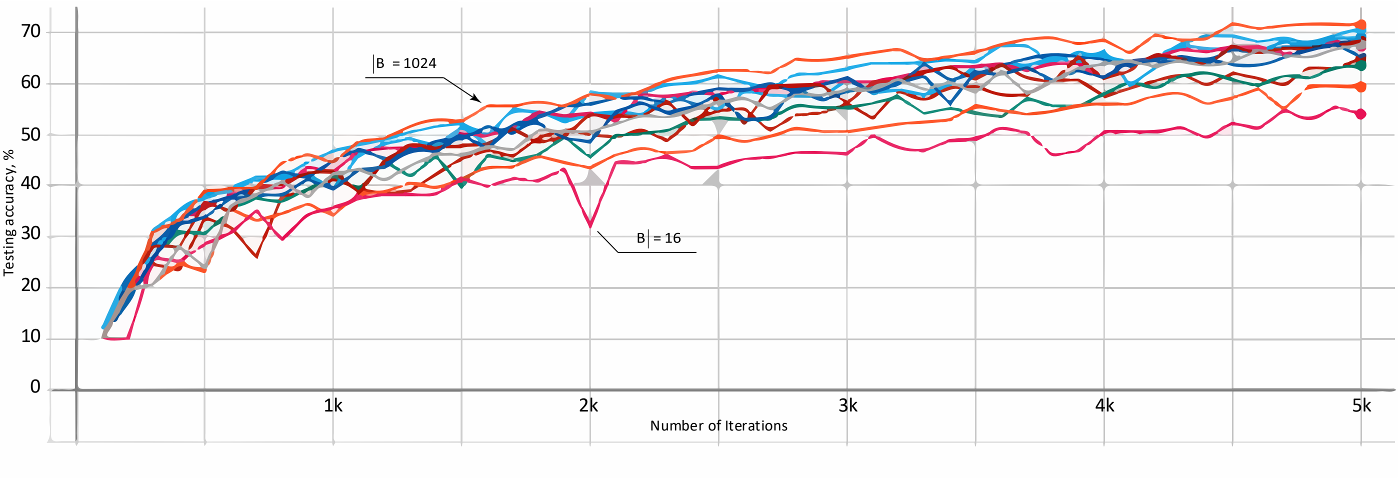

Fig. 4. The testing accuracy of the trained CNN with sequences B_k and B_l on the CIFAR-10 dataset. The smoothness of the curves is approximatly the same for all batch size values. The lowest curve corresponds to the batch size of 16 examples, the highest one – to 1024 examples.

TABLE I

THE FINAL TESTING ACCURACY RESULTS

$\|B\|$	Testing accuracy, %		$\|B\|$	Testing accuracy, %	
	MNIST	CIFAR-10		MNIST	CIFAR-10
16	97.42	54.05	150	98.58	68.33
32	97.78	59.38	200	98.84	68.71
50	98.44	63.59	250	98.70	67.32
64	98.39	64.96	256	98.22	68.97
100	98.75	67.67	512	98.93	70,80
128	98.51	65.24	1024	99.11	71.53

First of all, it should be noted that the batch size change trend is similar for the both considered datasets. The worst values of the test accuracy are demonstrated by the batch size of 16, 32, 50 and 64 examples. The best results of recognition accuracy are obtained from the batch size of 512 and 1024 examples. The batch sizes of 100, 128, 150, 200, 250 and 256 examples represent the average result of testing accuracy. Hence, the larger the batch size value, the higher the image recognition accuracy. Similar average batch size values from sequences B_k and B_l were compared and displayed below, in Figs. 5–7.

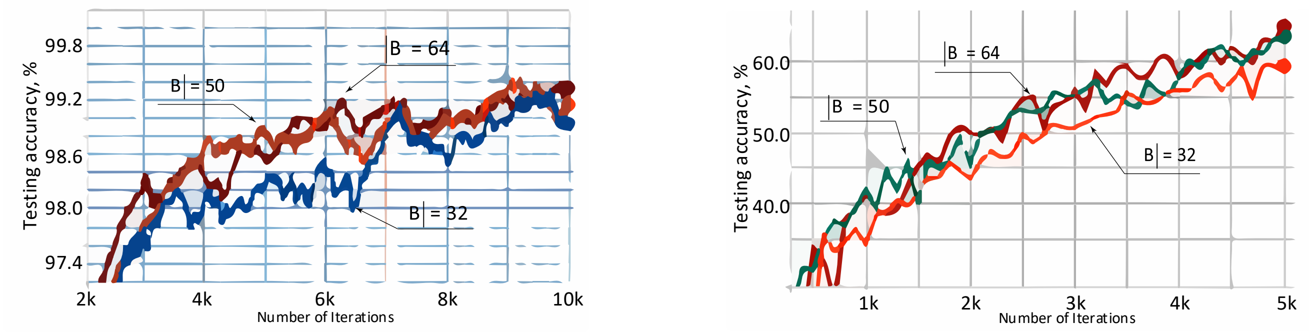

Fig. 5. The testing accuracy of the batch size values of 32, 50 and 64 on the MNIST (left figure) and CIFAR-10 (right figure) datasets. The best test accuracy result is demostrated by the batch size of 64 examples on the both datasets. The lowest testing accuracy corresponds to the batch size of 32 examples. The batch size with the value of 50 shows result, which is close to the value of 64.

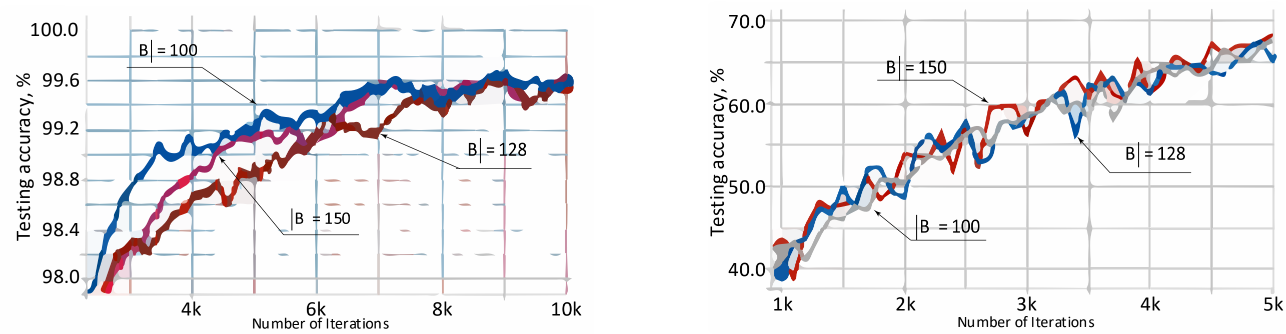

Fig. 6. The testing accuracy of the batch size values of 100, 128 and 150 on the MNIST (left figure) and CIFAR-10 (right figure) datasets. The trend of the curve growth is similar, however, a predominance of the batch size of 150 examples is observed.

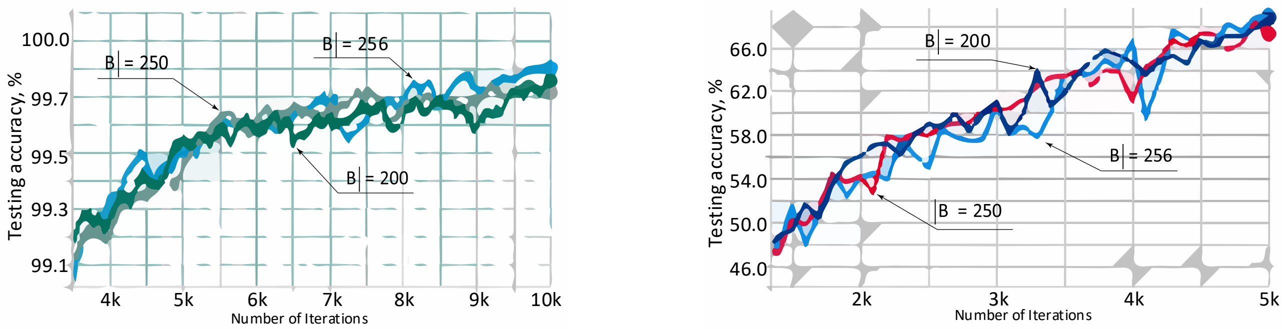

Fig. 7. The testing accuracy of the batch size values of 200, 250 and 256 on the MNIST (left figure) and CIFAR-10 (right figure) datasets. All three batch size values show almost identical result of testing accuracy on the both datasets. Nonetheless, on greater iterations, the batch size of 256 examples performes slightly better. Hense, at greater values of the batch size, the value of the test accuracy increases.

The training time efficiency is similar to the testing accuracy change trend for the MNIST and CIFAR-10 datasets: the higher the batch size value, the more time is required to train the network. The final time expenditures of training the network are shown in Table II.

TABLE II
THE TRAINING TIME EFFICIENCY

$\|B\|$	Time efficiency, h		$\|B\|$	Time efficiency, h	
	MNIST	CIFAR-10		MNIST	CIFAR-10
16	0.28	3.52	150	1.82	7.29
32	0.45	2.48	200	2.25	9.57
50	0.65	3.18	250	3.31	11.80
64	0.93	4.00	256	2.88	12.68
100	1.13	6.35	512	9.35	17.82
128	1.63	6.50	1024	14.23	27.47

As a result of the comparative analysis, the supposition about the dependence of the recognition accuracy on the batch size value was confirmed: the larger the batch size value, the higher the testing accuracy. Another supposition about the impact of the type of the batch size value on the CNN performance was not confirmed.

VII. Future Research

While this paper does show the impact of the batch size on the performance of CNN, depending on the parameter value, further research is needed to establish a more precise relationship between the training set size and the recognition accuracy. It is proposed to consider a combinatorial optimisation problem with an objective function as the recognition accuracy and an instance as the batch size parameter.

VIII. Conclusion

The problem of improving performance of CNNs is relevant and still not resolved. Adjusting CNN training parameters is one of the ways to accomplish this task. The training parameter of the batch size has been investigated in the article. The current investigation has shown that the batch size parameter has a crucial effect on the accuracy of image recognition. The greater the parameter value, the higher the image recognition accuracy. On the other hand, the large batch size value leads to huge computational costs.

The results of the research have not confirmed the assumption of impact of a certain type on the batch size value: neither numbers to the power of two nor numbers multiple of ten lead to a critical change in the recognition accuracy. Therefore, the optimal batch size varies from 200 and greater, depending on the computational resources.

Acknowledgement

The research has been supported by the Centre of Parallel Computations at Khmelnitsky National University, Ukraine.

References

[1] P. Goyal, P. Dollar, R. Girshick, P. Noordhuis, "Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour," *Facebook AI Research (FAIR), In CVPR*, 2017.
[2] L. Bottou. "Online Learning and Stochastic Approximations," *Online Learning and Neural Networks*, 1998.
[3] D. Mishkina, N. Sergievskiyb, J. Matasa, "Systematic Evaluation of CNN Advances on the ImageNet," *Center for Machine Perception, Faculty of Electrical Engineering*, 2016.
[4] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *In Proceedings of The 32nd International Conference on Machine Learning*, pp. 448–456, 2015.
[5] L. Wang, Y. Yang, R. Min, and S. Chakradhar, "Accelerating Deep Neural Network Training with Inconsistent Stochastic Gradient Descent," Neural Networks, vol. 93, pp. 219–229, Sep. 2017. https://doi.org/10.1016/j.neunet.2017.06.003
[6] M. Dereziński, D. Mahajan, S. S. Keerthi, S. V. N. Vishwanathan and M. Weimer, *Batch-Expansion Training: An Efficient Optimization Paradigm for Machine Learning*. 2017.
[7] N. Sh. Keskar, Dh. Mudigere, J. Nocedal, M. Smelyanskiy and P. T.-P. Tang, "On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima," *ICLR*, 2017.
[8] D. R. Wilson and T. R. Martinez, "The General Inefficiency of Batch Training for Gradient Descent Learning," *Neural Networks*, vol. 16, no. 10, pp. 1429–1451, Dec. 2003. https://doi.org/10.1016/s0893-6080(03)00138-2
[9] M. Takac, A. Bijral, P. Richtarik and N. Srebro, "Mini-Batch Primal and Dual Methods for SVMs," *JCMB*, 2013.
[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016. https://doi.org/10.1109/cvpr.2016.90
[11] A. Krizhevsky, "One Weird Trick for Parallelizing Convolutional Neural Networks," *In CoRR*, 2014.
[12] K. Simonyan and A. Zisserman "Very Deep Convolutional Networks for Large-Scale Image Recognition," *In Proceedings of ICLR*, 2015.
[13] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient Mini-Batch Training for Stochastic Optimization," *In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining – KDD '14*, 2014. http://dx.doi.org/10.1145/2623330.2623612
[14] Z. Lin, M. Courbariaux, R. Memisevic and Y. Bengio, "Neural networks with Few Multiplications," *In Proceedings of the 32d International Conference on Machine Learning, ICML '16*, pp. 561–568, 2016.
[15] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-Column Deep Neural Networks for Image Classification," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2012. https://doi.org/10.1109/cvpr.2012.6248110
[16] V. V. Romanuke, "Training Data Expansion and Boosting of Convolutional Neural Networks for Reducing the MNIST Dataset Error Rate," *Research Bulletin of the National Technical University of Ukraine "Kyiv Polytechnic Institute,"* vol. 0, no. 6, pp. 29–34, Dec. 2016. https://doi.org/10.20535/1810-0546.2016.6.84115
[17] V. V. Romanuke, "Appropriate Number and Allocation of RELUS in Convolutional Neural Networks," *Research Bulletin of the National Technical University of Ukraine "Kyiv Politechnic Institute,"* vol. 0, no. 1, pp. 69–78, Mar. 2017. https://doi.org/10.20535/1810-0546.2017.1.88156
[18] Y. LeCun and Y. Bengio, "Convolutional Networks for Images, Speech, and Time-Series," *The Handbook of Brain Theory and Neural Networks*, 1995.
[19] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. 2016.

Radiuk M. Pavlo graduated from Khmelnitsky National University (Ukraine) in 2017 and received the Master's degree in Mathematical and Computer Modelling. In 2017, Pavlo Radiuk became a Doctoral student at Khmelnitsky National University. His current research interests concern statistical analysis and machine learning.
Address for correspondence: 11 Institutskaya Street, Khmelnitskiy, Ukraine, 29016.
E-mail: radiukpavlo@gmail.com