УДК 004.896
DOI: 10.17586/0021-3454-2023-66-11-926-935

REVIEW ON OPTIMIZATION TECHNIQUES OF BINARY NEURAL NETWORKS

A. SHAKKOUF

ITMO University, St. Petersburg, Russia
ashakkuf@itmo.ru

Abstract. The deployment of Convolutional Neural Networks (CNNs) models on embedded systems faces multiple problems regarding computation power, power consumption and memory footprint. To solve these problems, a promising type of neural networks that uses 1-bit activations and weights emerged in 2016 called Binary Neural Networks (BNNs). BNN consumes less energy and computation power mainly because it replaces the complex heavy convolution operation with simple bitwise operations. However, the quantization from 32-float point to 1-bit leads to accuracy loss and poor performance, especially on large datasets. This article presents a review of the key optimization techniques which influenced the performance of BNNs and led to higher representation capacity of BNN models, as well as an overview of the application methods of BNNs in object detection tasks and compares the performance with the real value CNN.

Keywords: *binary neural networks, BNNs optimization, object detection, quantization, binarization, computer vision, artificial intelligence*

For citation: *Shakkouf A.* Review on Optimization Techniques of Binary Neural Networks. *Journal of Instrument Engineering.* 2023. Vol. 66, N 11. P. 926—935 (in English). DOI: 10.17586/0021-3454-2023-66-11-926-935.

ОБЗОР МЕТОДОВ ОПТИМИЗАЦИИ БИНАРНЫХ НЕЙРОННЫХ СЕТЕЙ

А. Шаккуф

Университет ИТМО, Санкт-Петербург, Россия
ashakkuf@itmo.ru

Аннотация. Развертывание моделей сверточных нейронных сетей (СНС) во встраиваемых системах осложнено множеством проблем, связанных с вычислительной мощностью, энергопотреблением и объемом памяти. Для решения этих проблем в 2016 г. создан многообещающий тип нейронных сетей, использующих 1-битную активацию и веса, — бинарные нейронные сети (БНС). Такие сети потребляют меньше энергии и вычислительных мощностей, так как заменяют сложную операцию тяжелой свертки простыми побитовыми операциями. Однако квантование с 32-разрядной плавающей запятой до 1 бита приводит к потере точности и снижению производительности, особенно при больших наборах данных. Представлен обзор ключевых методов оптимизации, которые повлияли на производительность БНС и привели к повышению репрезентативности их моделей, также представлены обзор способов применения БНС в задачах обнаружения объектов и сравнительный анализ их производительности с реальным значением.

Ключевые слова: бинарные нейронные сети, оптимизация БНС, обнаружение объектов, квантование, бинаризация, компьютерное зрение, искусственный интеллект

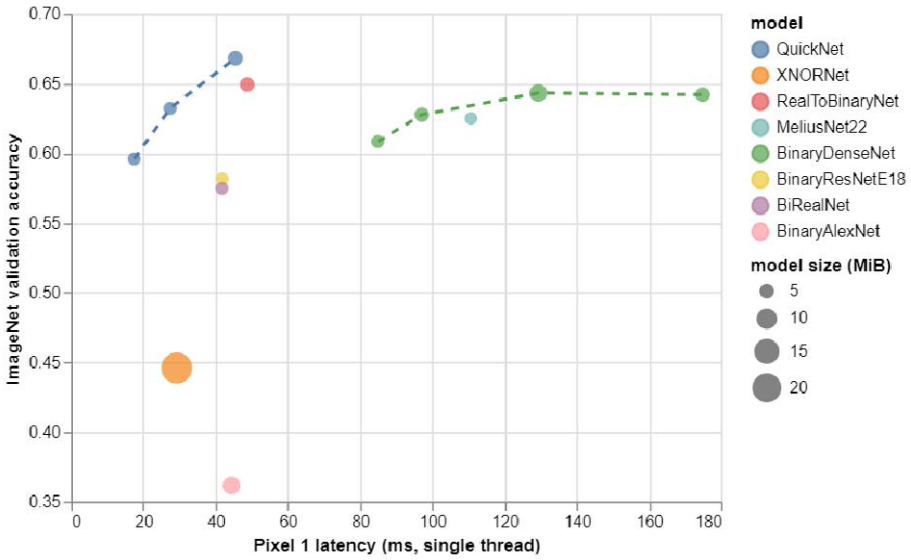
Ссылка для цитирования: Шаккуф А. Обзор методов оптимизации бинарных нейронных сетей // Изв. вузов. Приборостроение. 2023. Т. 66, № 11. С. 926—935. DOI: 10.17586/0021-3454-2023-66-11-926-935.

Introduction. Convolutional Neural Networks (CNN) have pushed Artificial Intelligence (AI) limits in many aspects, including but not limited to image classification [1, 2], object recognition [3, 4], speech emotion recognition [4—6], object detection [7] and classification of noisy signals [8]. CNNs have heavy designs with massive computational costs and parameters size, which makes it difficult to deploy CNN on the edge and portable devices without model compressing techniques. One of compression techniques is quantization, in which network parameters are represented with data types of smaller size. The most severe quantization technique in binarization, in which weights and activations are represented using 1-bit and the resulting networks are called Binary Neural Net-

works (BNNs). BNNs represent the ideal class of neural network for edge inference especially for battery driven devices, due to their use of XNOR for multiplication: a fast and cheap operation to perform with much smaller times of memory accesses. Times of memory access is important because each hardware consumes certain amount of energy for each memory access [9]. Moreover, their parameters are 32x times more compact, which increases opportunities for caching, providing further potential performance boosts. However, binarization dramatically improves inference speed but accuracy is greatly affected. For example, binary connect network performs classification on CIFAR-10 dataset with accuracy 10% less than the accuracy of the real value network [10] and the loss in accuracy in much larger on largescale datasets such as ImageNet. Figure* shows the great benefits of some BNN models in terms of models' sizes with acceptable inference latency. The loss in representation capacity of BNNs makes research for better binary feature maps representation -while training- a matter of central importance. Because of that, starting from 2016, a lot of research has been done to optimize BNNs and test its' performance in real applications. There are few reviews on BNNs, but our review is different from all other reviews in two points:

— we summarize the key optimization techniques that improved the performance of BNN to a large extent; other reviews summarize all the previously done research;

— we focus on works that use BNNs in object detection tasks and review all the previously conducted research in this field of computer vision.



Key optimization techniques of BNNs. Optimizing the training process of BNN is essential to gain the availability to train BNN on the edge. [11] provides a new low-cost strategy for BNN training that reduces the used memory by up to 5.44x while inducing little to no accuracy loss. Authors notice that high-precision activations should not be used while training BNNs, since we are only concerned with weights and activations' signs. Specifically, authors of [11] present the first successful combination of binary activations and binary and binary weight gradients during neural network training. An intuitive method to lower the memory footprint of training is to simply reduce the batch size. However, doing so generally leads to increased total training time due to reduced memory reuse [12]. The method in [11] does not conflict with batch size tuning, and further allows the use of large batches while remaining within the memory limits of edge devices. Authors used the standard BNNs training method of Courbariaux [10] as a baseline for comparison.

Authors in [10] introduce a method for training BNNs and perform two sets of experiments on two platforms: Torch7 and Theano. They operate on the binarizations approaches introduced by

* Larq implementation of deep neural networks with extremely low precision weights and activations.

[13]. [13] introduces two approaches to transfer high-precision NNs to BNNs. The first approach is deterministic and the other one is stochastic. The deterministic approach is formulated as:

$$x^b = \operatorname{sgn}(x) = \begin{cases} +1 & \text{if } x \geq 0, \\ -1 & \text{otherwise.} \end{cases}$$

Where x^b is the binarized value (weight or activation), x is the high-precision variable. While the stochastic approach is formulated as:

$$x^b = \begin{cases} +1 & \text{with probability } p = \sigma(x); \\ -1 & \text{with probability } 1-p, \end{cases}$$

where σ is the hard sigmoid function; $\sigma(x) = \operatorname{clip}\left(\frac{x+1}{2}, 0, 1\right) = \max\left(0, \min\left(1, \frac{x+1}{2}\right)\right)$.

Courbariaux [10] states that stochastic binarization is more appealing than the deterministic one, but harder to implement as it requires the hardware to have a random generation unit (peripheral). So, it is quite often preferable to use the deterministic approach over the stochastic one. The negative side of [10] is that real-valued gradients of the weights are accumulated in real-valued variables because they are required for Stochastic Gradient Descent (SGD) to work at all. However, this problem has been solved by [11]. The derivative of sign function is zero almost everywhere, and that prevent performing backpropagation. This problem has been solved by [14, 15], where the authors introduced what is called "straight-through estimator". [10] uses the same approach as [14, 15] for gradient estimation but adds to it a saturation effect. Authors in [10] also wrote an optimized binary matrix multiplication kernel for GPU which performs 7x faster than the unoptimized GPU kernel.

It is well known that we add a regularization term like L_1 and L_2 to a model to prevent overfitting and as a result we obtain robust generalization. If we use these regularization functions while training binary NN, it will direct the weights to be near zero and this is not compatible with BNNs, because we need the weights to be around –1 and +1. So, to make the regularization term more general, authors in [16] introduces scaling factor α which makes the regularization function symmetric and has two minimums at $-\alpha$, $+\alpha$. Those scales are embedded into the layers parameters and thus are learnable while training.

Authors in [17] provide a smart algorithm (framework) for automatic search of compact but accurate BNNs architecture. The main idea is to expand — while we binarize the network — each layer of the network by a factor of a where $a \in \{0.25,\ 0.5,\ 1,\ 2,\ 3,\ 4\}$. Specifically, authors create a generation of networks architectures, each architecture corresponds to an expansion ratio. After training, we choose the best candidate (best BNN architecture) using a fitness function $f\left(a^k\right) = \max\left(\text{Acc} - \lambda \times \text{FLOPs}, 0\right)$ Where FLOPs and Acc are float operations and Top-1 validation accuracy of the network of an individual a^k, λ is the trade-off parameter.

Applying x-nor and bit-count operations causes and accumulates notable quantization error, which usually results in inconsistent signs in binary feature maps compared with their full-precision counterpart [18]. To handle this inconsistency, [18] present a channel-wise interaction based binary convolutional neural network learning method (CI-BCNN) to learn BNN with channel-wise interactions to reduce the accumulated error and obtain an efficient inference. While in [19], authors approximate the real value weights with linear combination of multiple binary bases and use that to alleviate information loss in the forward pass. A network called Bi-Real proposed in [20] connects the float activations to activations of the consecutive block, through an identity shortcut. Conse-

quently, compared to the standard 1-bit CNN, the representational capability of the Bi-Real net is significantly enhanced.

Unlike [18], [21] proposes an approach that gives weights to binaries variables and is called Balanced Binary neural networks with Gated residual (BBG for short). First, weight-balanced binarization is presented so binary weights can capture more information contained in activations. Second, a gated residue is appended to make recompense for the loss of information during the forward pass, with a slight increase. Both techniques can be encapsulated as a generic network module that supports different network architectures for different tasks including detections. Authors assure deployment efficiency on mobile devices using a framework called daBNN and was introduced by [22].

According to central limit theorem (CLT) [23, 24], the general description for activation is that they are nearly Gaussian, which makes it hard for the $sign(.)$ function to capture the higher-order statistics such as variance. This fact motivated the authors of [25] to propose a new approach for binarization called „Sparsity-Inducing BNN“ (Si-BNN). The new approach tries to maximize the mutual information between inputs and outputs of a single layer by a proper (optimal) choosing of the sparsity threshold θ. Binarization equation and backward gradient estimation via straight-through estimator (STE) formulas are:

$$X_b = \varphi(X) = \begin{cases} +1 & \text{if } x \geq \theta, \\ 0 & \text{otherwise;} \end{cases} \quad \frac{\partial \varphi}{\partial X} = \begin{cases} +1 & \text{if } 0 \geq X \geq +1, \\ 0 & \text{otherwise.} \end{cases}$$

Training Si-BNN and testing on ImageNet, MNIST and FICAR-10 benchmarks demonstrate that Si-BNN dramatically outperforms current best performing methods like QNet in [26] and BENN-6, Bagging in [27], lowering the performance gap between full-precision networks and binarized neural networks.

Compared to previous research that demonstrated the viability of BNNs via experiments, [28] explains why these BNNs work in terms of the High-Dimensional geometry. [28] shows that BNNs trained using the method of [10] work because of the high-dimensional geometry of binary vectors. In particular, the ideal continuous vectors that extract out features in the intermediate representations of these BNNs are well-approximated by binary vectors in the sense that dot products are approximately preserved. This theory serves as a foundation for understanding not only BNNs but a variety of methods that seek to compress traditional NNs using the well-known compression techniques mentioned in [29, 30]. A promising technique to enhance BNNs representation capacity introduced in [31] where authors refined the kernel and features using generative adversarial learning like KR-GAL and FR-GAL. [32] empirically proves that quantizing the weights can improve generalization, where authors show that eigenvalue of neural tangent kernel of the proposed network decays approximately exponentially.

Application of BNNs in Object Detection. Authors of [33] noticed that binarization leads to poor representation capabilities of features. To avoid that [34] proposes a method called “Block Scaling Factor XNOR” (BSF-XNOR). This method is built on the XNOR binarization algorithm [35] but adds to it better representation capabilities using a scaling factor for each block under a used filter and increasing in operation parallelization without increasing the calculation amount. The scaling factor is calculated using a specific mathematical expression introduced by [34]. The suggested algorithm was applied on unmanned aerial vehicles (drones). BSF-XNOR beats most of the well know algorithms for object detection in overall performance like XNOR, YOLOv3-tiny, Non-bin, XYOLO [36].

To simplify the search for appropriate architecture of BNN, [37] proposes an algorithm called BNAS which produces high compact models for detection tasks. [38] presents a method for object detection in infrared images using BNNs. The authors demonstrate that the perfor-

mance of BNNs is very close to that of 32-bit floating-point networks on the IR dataset and present a system architecture (using external DRAM an internal SRAM) designed specifically for computation using binary representation. [38] shows that BNNs can achieve high recognition accuracy while reducing memory and energy requirements, making them suitable for use in embedded platforms and mobile devices. [39] proposes a new approach for object detection using a fast unified binary network. The proposed method is based on the X-NOR network and uses binary-precision convolution. The network also uses convolution kernels of different sizes to predict classes and bounding boxes of multi-scale objects directly which makes the approach easy to implement in embedded computing systems and achieves faster object detection with acceptable loss of accuracy.

A modified binarized convolutional neural network proposed in [40] can reduce power consumption without any speed loss and improve system performance while keeping low power dissipation. The article also describes the limitations of reducing power consumption through software and how optimized SoC hardware structure can extend the limitation of software methods, for example, by the use AXI interfaces to accelerate the process and optimize data transferring.

Authors in [41] propose a low bit-width weight optimization approach to train BNN called (BDNN). This method uses a greedy layer-wise technique to train the detection network instead of binarizing the whole network once at a time, which boosts performance instead of training the entire network at the same time.

To optimize the detection process for time, [42] introduces a point-process filter (PPF) that filters the input video stream to remove the noise. After that, the filtered images are passed to an efficiently implemented BNN on FPGA. The implementation shows a reduction of 86% in latency compared to the full precision NN.

[43] proposes a binarized neural network learning method called BiDet for efficient object detection. This method eliminates the redundant information using the principle of information bottleneck which gives us a fully utilization of the representational capacity of the networks and enforces the posteriors to be concentrated on informative prediction for false positive elimination, through which the detection precision is significantly enhanced.

To maintain a performance so close to that of real value NN, [44] presents a strategy called layer-wise searching which generates 1-bit detectors that minimize the angular error in a student-teacher framework. To increase the capacity of the detectors, authors introduce angular and amplitude loss functions. Those functions search learns the scale factor that minimizes the amplitude error and finds the optimal binary weights that minimize angular loss. On the other hand, authors in [45] try to increase features representation capacities by using an adaptive amplitude method that reformulates the binary convolution. A good comparison of the performance of different NNs in detection tasks was carried out by [46], where they compared previously trained CNN, QNN and BNN. The detection of small objects manipulated by hand was studied in [47] for surveillance purposes, where the authors implemented robust and reliable model for detection based on binarization techniques. A very actual detection task was studied by [48] which used BNN (DAD-Net) to detect drivable areas (segmentation) for autonomous driving which saves energy and computing power. The proposed network uses binary weights and activations in both encoder and encoder parts and in the bottleneck. To keep passengers safe in public transportation and alert for anomaly state, [49] implemented a BNN for faster emotion recognition from facial expressions. [50] performs semantic segmentation through GroupNet algorithm. GroupNet divides the network into sub-groups and performs approximation for each sub-group using combinations of binary bases. Table illustrates the BNN results of the object detection task on the benchmark datasets PASCAL VOC.

Summary of BNNs performance on object detection for PASCAL VOC dataset

Neural Network Approach	Network Architecture	Binarization method / Real-valued	Trained Dataset	mAP%
Customized	VGG16	Real-valued	VOC2007	68.9
		BNN	VOC2007	47.3
	Alexnet	Real-valued	VOC2007	66.0
		BNN	VOC2007	46.4
Faster RCNN	VGG16	BDNN	VOC2012	62.6
	ResNet-18	Real-valued	VOC2007	67.8
		Bi-Real Net	VOC2007	51.0
	ResNet-18	Real-valued	VOC2007+2012	73.2
		Bi-Real Net	VOC2007+2012	60.6
	ResNet-34	Real-valued	VOC2007+2012	75.6
		XNOR-Net	VOC2007+2012	54.7
	ResNet-18	Real-valued	VOC2007+2012	74.5
		BiDet	VOC2007+2012	50.0
		BiDet(SC)		59.5
		XNOR-Net		48.4
		Bi-Real Net		58.2
	ResNet-18	Real-valued	VOC2007+2012	76.4
		Bi-Real Net	VOC2007+2012	60.9
		BiDet		62.7
	ResNet-34	Real-valued	VOC2007+2012	77.8
		Bi-Real Net	VOC2007+2012	63.1
		BiDet		65.8
	ResNet-50	Real-valued	VOC2007+2012	79.5
		Bi-Real Net	VOC2007+2012	65.7
	ResNet-18	Real-valued	VOC2007	74.5
		DA-BNN	VOC2007	63.5
YOLOv2	DarkNet	XNOR-Net	VOC2007	79.6
SSD	VGG16	BDNN	VOC2007+2012	63.3
		XNOR-Net		60.71
SSD300	VGG16	Real-valued	VOC2007+2012	72.4
		BiDet	VOC2007+2012	52.4
		BiDet(SC)		66.0
		XNOR-Net		50.2
		Bi-Real Net		63.8
	MobileNetV1	Real-valued	VOC2007+2012	68.0
		BiDet	VOC2007+2012	51.2
		XNOR-Net		48.9
	VGG16	Real-valued	VOC2007+2012	74.3
		Bi-Real Net	VOC2007+2012	63.8
		BiDet		66.0

Conclusion. Although BNNs have some aspects to be used in, a few challenges and constraints remain an open issue for research. For a given task, what is the architecture of BNN we should use? In general, all the layers (except the input and output layers) of a BNN are binarized CNN layers, and this is a primary source for information loss. The deeper the BNN the more we lose information because the performance drop is accumulated from the previous layers. In this paper, we conducted a review on the key optimization techniques for BNN (training strategies, binarization methods, increasing representation capacity) and a review of the application and real-life tasks that used BNNs to handle object detections.

REFERENCES

1. Basha S.S., Dubey S.R., Pulabaigari V., and Mukherjee S. *Neurocomputing*, 2020, vol. 378, pp. 112–119.
2. Zhou W., Wang H., and Wan Z. *Computers and Electrical Engineering*, 2022, vol. 99, art. no. 107819.
3. Gao X., Xing G., Roy S., and Liu H. *RIEEE Sensors J.*, 2021, no. 4(21), pp. 5119–5132.
4. Ashiq F., Asif M., Ahmad M. B., Zafar S., Masood K., Mahmood T., Mahmood M.T., and Lee I.H. *IEEE Access*, 2022, vol. 10, pp. 14819–14834.
5. Abdelhamid A.A., El-Kenawy E.-S.-M., Alotaibi B., Amer G.M., Abdelkader M.Y., Ibrahim A., and Eid M.M. *IEEE Access.*, 2022, vol. 10, pp. 49265–49284.
6. Kwon S. *Expert Systems with Applications*, 2021, vol. 167, art. no. 114177.
7. Zhang N., Wei X., Chen H., and Liu W. *Electronics*, 2021, vol. 10, no. 3, p. 282.
8. Lopac N., Hrzic F., Vuksanovic I.P., and Lerga J. *IEEE Access.*, 2022, vol. 10, pp. 2408–2428.
9. Horowitz M. *IEEE Intern. Solid State Circuits Conf.*, 2014, pp. 10–14.
10. Courbariaux M. and Bengio Y. *ArXiv journal,* 2016, preprint arXiv: 1602.02830.
11. Wang E., Davis J.J., Moro D., Zielinski P., Lim J.J., Coelho C., Chatterjee S., Cheung P.Y., Constantinides G.A. *ACM Transactions on Embedded Computing Systems*, 2021, DOI:10.1145/3626100.
12. Sohoni N.S., Aberger C.R., Leszczynski M., Zhang J., and Ré C. *ArXiv journal*, 2019, preprint arXiv:1904.10631.
13. Courbariaux M., Bengio Y., and David J.-P. *ArXiv e-prints*, 2015, abs/1511.00363.
14. Hinton G. *Neural networks for machine learning*, Coursera, video lectures, 2012.
15. Bengio Y. *Technical Report,* Universite de Montreal, arXiv:1305.2982, 2013.
16. Darabi S., Belbahri M., Courbariaux M., and Nia V.P. *BNN+: Improved binary network training*, 2019, https://openreview.net/pdf?id=SJfHg2A5tQ.
17. Shen M., Han K., Xu C., Wang Y. *IEEE/CVF Intern. Conf. on Computer Vision Workshops*, 2019.
18. Wang Z., Lu J., Tao C., Zhou J., Tian Q. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, vol. 568.
19. Lin X., Zhao C., and Pan W. *NIPS*, 2017, pp. 344–352.
20. Liu Z., Wu B., Luo W., Yang X., Liu W., and Cheng K.-T. *arXiv preprint*, 2018, arXiv:1808.00278.
21. Shen M., Liu X., Gong R., Han K. *IEEE Intern. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, vol. 4197.
22. Jianhao Zhang, Yingwei Pan, Ting Yao, He Zhao, and Tao Mei. *arXiv preprint*, 2019, arXiv:1908.05858.
23. Cai Z., He X., Sun J., and Vasconcelos N. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.
24. Ioffe S., and Szegedy C. *Proc. of the 32nd Intern. Conf. on Machine Learning, ICML*, 2015, pp. 448–456.
25. Wang P., He X., Li G., Zhao T., Cheng J. *AAAI Conf. on Artificial Intelligence*, 2020, vol.34, pp. 12192.
26. Yang J., Shen X., Xing J., Tian X., Li H., Deng B., Huang J., and Hua X.-S. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
27. Zhu S., Dong X., and Su H. *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.
28. Anderson A.G. & Berg C.P. *ArXiv journal*, 2017, abs/1705.07199.
29. Qin H., Gong R., Liu X., Bai X., Song J., Sebe N. *Pattern Recognition*, 2020, 105. 107281. 10.1016/ j.patcog.107281.
30. Liang T., Glossner J., Wang L., Shi S., and Zhang X. *Neurocomputing*, 2021, vol. 461, pp. 370–403.
31. Xu Sh., Liu Ch., Zhang B., Lu J., Guo G, Doermann D. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2022, 18. 10.1145/3473340.
32. Zhang K., Yin M., & Wang Y. *ArXiv journal*, 2022, abs/2206.05916.
33. Darabi S., Belbahri M., Courbariaux M., and Nia V.P. *ArXiv journal*, 2018, 1812.11800.
34. Wang S., Zhang C., Su D., Wang L., Jiang H. *IEEE Access 9*, 2021, pp. 106169.
35. Rastegari M, Ordonez V, Redmon J, and Farhadi A. *ECCV*, 2016, pp. 525– 542.
36. Barry D., Shah M., Keijsers M., Khan H., and Hopman B. *ArXiv journal: 1910.03159*, 2019.
37. Chen H., Zhuo L., Zhang B., Zheng X., Liu J., Ji R., Doermann D., Guo G. *ArXiv journal*, 2020.
38. Kung J., Zhang D., van der Wal G., Chai S., Mukhopadhyay S. *Journal of Signal Processing Systems*, 2018, vol. 90, pp. 1–14, DOI: 10.1007/s11265-017-1255-5.
39. Wang X., Siyang S., Yin Y., Xu D. & Wu W., Gu Q. *CAAI Transactions on Intelligence Technology*, 2018, no. 3(4), DOI:10.1049/trit.2018.1026.
40. Kim H. and Choi K. *Intern. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData)*, 2019, pp. 240–243.
41. Peng H. and Chen S. *Pattern Recognit. Lett.*, 2019, vol. 125, pp. 91–97.
42. Ojeda F.C., Bisulco A., Kepple D., Isler V., and Lee D.D. *Intern. Conf. Image Process. (ICIP)*, 2020, pp. 3084–3088.
43. Wang Z., Wu Z., Lu J., and Zhou J. *IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2049–2058.
44. Xu S., Zhao J., Lu J., Zhang B., Han S., and Doermann D. *IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2021, pp. 5682–5691.
45. Zhao J., Xu S., Wang R., Zhang B., Guo G., Doermann D., and Sun D. *Pattern Recognit. Lett.*, 2022, vol. 153, pp. 239–245.
46. Mani V.R.S., Saravanaselvan A., and Arumugam N. *J. Microelectron.*, 2022, vol. 119, art. no. 105319.
47. Pérez-Hernández F., Tabik S., Lamas A., Olmos R., Fujita H., Herrera F. *Knowl. Base Syst.*, 2020, art. no.105590, https://doi.org/10.1016/j.knosys.2020.105590.
48. Frickenstein A., Vemparala M.-R., Mayr J., Nagaraja N.-S., Unger C., Tombari F., and Stechele W. *IEEE Intern. Conf. Robot. Autom. (ICRA)*, 2020, pp. 2295–2301.
49. Ajay B.S. and Rao M. *34th Intern. Conf. VLSI Design; 20th Intern. Conf. Embedded Syst. (VLSID)*, 2021, pp. 175–180.
50. Zhuang B., Shen C., Tan M., Liu L., and Reid I. *IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 413–422.

Data on author

Ali Shakkouf — Post-Graduate student; ITMO University, Faculty of Control Systems and Robotics; E-mail: ashakkuf@itmo.ru

Received 05.06.2023; approved after reviewing 22.06.2023; accepted for publication 27.09.2023.

СПИСОК ЛИТЕРАТУРЫ

1. *Basha S. S., Dubey S. R., Pulabaigari V. and Mukherjee S.* Impact of fully connected layers on performance of convolutional neural networks for image classification // Neurocomputing. 2020. Vol. 378. P. 112—119.
2. *Zhou W., Wang H. and Wan Z.* Ore image classification based on improved CNN // Computers and Electrical Engineering. 2022. Vol. 99, art. N 107819.
3. *Gao X., Xing G., Roy S., and LiuH.* RAMP-CNN: A novel neural network for enhanced automotive radar object recognition // IEEE Sensors J. 2021. Vol. 21, N 4. P. 5119—5132.
4. *Ashiq F., Asif M., Ahmad M. B., Zafar S., Masood K., Mahmood T., Mahmood M. T. and LeeI. H.* CNN-based object recognition and tracking system to assist visually impaired people // IEEE Access. 2022. Vol. 10. P. 14819—14834.
5. *Abdelhamid A. A., El-Kenawy E.-S.-M., Alotaibi B., Amer G. M., Abdelkader M. Y., Ibrahim A. and Eid M. M.* Robust speech emotion recognition using CNN+LSTM based on stochastic fractal search optimization algorithm // IEEE Access. 2022. Vol. 10. P. 49265—49284.
6. *Kwon S.* MLT-DNet: Speech emotion recognition using 1D dilated CNN based on multi-learning trick approach // Expert Systems with Applications. 2021. Vol. 167, art. N 114177.
7. *Zhang N., Wei X., Chen H. and Liu W.* FPGA implementation for CNN-based optical remote sensing object detection // Electronics. 2021. Vol. 10, N 3. P. 282.
8. *Lopac N., Hrzic F., Vuksanovic I. P. and Lerga J.* Detection of non-stationary GW signals in high noise from Cohen's class of time–frequency representations using deep learning // IEEE Access. 2022. Vol. 10. P. 2408—2428.
9. *Horowitz M.* Computing's Energy Problem (and what we can do about it) // IEEE Intern. Solid State Circuits Conf. 2014. P. 10—14.
10. *Courbariaux M. and BengioY.* BinaryNet: Training deep neural networks with weights and activations constrained to +1 or -1 // ArXiv Journal. 2016. preprint arXiv: 1602.02830.
11. *Wang E., Davis J. J., Moro D., Zielinski P., Lim J. J., Coelho C., Chatterjee S., Cheung P. Y., Constantinides G. A.* Enabling binary neural network training on the edge // 5th Intern. Workshop on Embedded and Mobile Deep Learning. 2021. June. P. 37—38.
12. *Sohoni N. S., Aberger C. R., Leszczynski M., Zhang J. and CRé.* Low-memory neural network training: A technical report // ArXiv journal. 2019. preprint arXiv:1904.10631.
13. *Courbariaux M., Bengio Y. and Jean-Pierre D.* Binaryconnect: Training deep neural networks with binary weights during propagations // ArXiv e-prints. 2015. abs/1511.00363.
14. *Hinton G*. Neural networks for machine learning // Coursera (video lectures). 2012.
15. *Bengio Yo*. Estimating or propagating gradients through stochastic neurons // Technical Report arXiv:1305.2982. Universite de Montreal. 2013.
16. *Darabi S., Belbahri M., Courbariaux M. and Nia V. P.* BNN+: Improved binary network training. 2019. https://openreview.net/pdf?id=SJfHg2A5tQ.
17. *Shen M., Han K., Xu C., Wang Y*. Searching for accurate binary neural architectures // IEEE/CVF Intern. Conf. on Computer Vision Workshops. 2019.
18. *Wang Z., Lu J., Tao C., Zhou J., Tian Q*. Learning channel-wise interactions for binary convolutional neural networks // IEEE/CVF Conf. on Computer Vision and Pattern Recognition. 2019. P. 568.
19. *Xiaofan Lin, Cong Zhao, and Wei Pan*. Towards accurate binary convolutional neural network // NIPS. 2017. P. 344—352.

20. *Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng*. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm // arXiv preprint. 2018. arXiv:1808.00278.

21. *Shen M., Liu X., Gong R., Han K*. Balanced binary neural networks with gated residual // IEEE Intern. Conf. on Acoustics, Speech and Signal Processing (ICASSP) 2020. 4197.

22. *Jianhao Zhang, Yingwei Pan, Ting Yao, He Zhao and Tao Mei*. dabnn: A super fast inference framework for binary neural networks on arm devices // arXiv preprint. 2019 arXiv:1908.05858.

23. *Cai Z.; He X.; Sun J. and Vasconcelos N*. Deep learning with low precision by half-wave gaussian quantization // IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2017.

24. *Ioffe S. and Szegedy C*. Batch normalization: Accelerating deep network training by reducing internal covariate shift // Proc. of the 32nd Intern. Conf. on Machine Learning, ICML. 2015. P. 448—456.

25. *Wang P., He X., Li G., Zhao T., Cheng J*. Sparsity-inducing binarized neural networks // AAAI Conf. on Artificial Intelligence. 2020. N 34. P. 12192.

26. *Yang J., Shen X., Xing J., Tian X., Li H., Deng B., Huang J. and Hua X.-s*. Quantization networks // IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2019.

27. *Zhu S., Dong X. and Su H*. Binary ensemble neural network: More bits per network or more networks per bit? // IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2019.

28. *Anderson A. G. and Berg C. P*. The High-Dimensional Geometry of Binary Neural Networks // ArXiv Journal. 2017. abs/1705.07199.

29. *Qin Haotong, Gong Ruihao, Liu Xianglong, Bai Xiao, Song Jingkuan, Sebe Nicu.* Binary Neural Networks: A Survey // Pattern Recognition. 2020. N 105. 107281. DOI: 10.1016/ j.patcog.107281.

30. *Liang T., Glossner J., Wang L., Shi S. and Zhang X.* Pruning and quantization for deep neural network acceleration: A survey // Neurocomputing. 2021. Vol. 461. P. 370—403.

31. *Xu Sheng, Liu Chang, Zhang Baochang, Lu Jinhu, Guo Guodong, Doermann D*. BiRe-ID: Binary Neural Network for Efficient Person Re-ID // ACM Trans. on Multimedia Computing, Communications, and Applications. 2022. N 18. DOI: 10.1145/3473340.

32. *Zhang K., Yin M. and Wang Y*. Why Quantization Improves Generalization: NTK of Binary Weight Neural Networks // ArXiv Journal. 2022. abs/2206.05916.

33. *Darabi S., Belbahri M., Courbariaux M. and Nia V. P.* Regularized binary network training // ArXiv Journal. 2018, 1812.11800.

34. *Wang S., Zhang C., Su D., Wang L., Jiang H*. High-precision binary object detector based on a bsf-xnor convolutional layer // IEEE Access 9. 2021. P. 106169.

35. *Rastegari M., Ordonez V., Redmon J. and Farhadi A*. Xnor-net: Imagenet classification using binary convolutional neural networks // ECCV. 2016. P. 525— 542.

36. *Barry D., Shah M., Keijsers M., Khan H. and Hopman B*. XYOLO: A model for real-time object detection in humanoid soccer on low-end hardware // ArXiv Journal: 1910.03159. 2019.

37. *Chen Hanlin, Zhuo Li'an, Zhang Baochang, Zheng Xiawu, Liu Jianzhuang, Ji Rongrong, Doermann D., Guo Guodong*. Binarized Neural Architecture Search for Efficient Object Recognition // ArXiv Journal. 2020.

38. *Kung Jaeha, Zhang David, van der Wal G. Chai, Sek Mukhopadhyay S*. Efficient Object Detection Using Embedded Binarized Neural Networks // Journal of Signal Processing Systems. 2018. N 90. P. 1—14. DOI: 10.1007/s11265-017-1255-5.

39. *Wang Xingang, Siyang Sun, Yin Yingjie, Xu De, Wu Wenqi, Gu Qingyi*. Fast Object Detection Based on Binary Deep Convolution Neural Networks // CAAI Trans. on Intelligence Technology. 2018. N 3. DOI: 10.1049/trit.1026.

40. *Kim H., Choi K.* The implementation of a power efficient BCNN based object detection acceleration on a Xilinx FPGA-SoC // Intern. Conf. Internet Things (iThings) IEEE Green Comput. Commun. (GreenCom) IEEE Cyber, Phys. Social Comput. (CPSCom) IEEE Smart Data (SmartData). 2019. P. 240—243.

41. *Peng H., Chen S.* BDNN: Binary convolution neural networks for fast object detection // Pattern Recognition Lett. 2019. Vol. 125. P. 91—97.

42. *Ojeda F. C., Bisulco A., Kepple D., Isler V. and Lee D. D.* On-device event filtering with binary neural networks for pedestrian detection using neuromorphic vision sensors // IEEE Intern. Conf. Image Process. (ICIP) 2020. P. 3084—3088.

43. *Wang Z., Wu Z., Lu J. and Zhou J.* BiDet: An efficient binarized object detector // IEEE/CVF Conf. Comput. Vis. Pattern Recognition. 2020. P. 2049—2058.

44. *Xu S., Zhao J., Lu J., Zhang B., Han S. and Doermann D.* Layer-wise searching for 1-bit detectors // IEEE/CVF Conf. Comput. Vis. Pattern Recognition. (CVPR). 2021. P. 5682—5691.

45. *Zhao J., Xu S., Wang R., Zhang B., Guo G., Doermann D. and Sun D.* Data-adaptive binary neural networks for efficient object detection and recognition // Pattern Recognition. Lett. 2022. Vol. 153. P. 239—245.

46. *Mani V. R. S., Saravanaselvan A. and Arumugam N.* Performance comparison of CNN, QNN and BNN deep neural networks for real-time object detection using Zynq FPGA node // Microelectron. 2022. Vol. 119, art. N 105319.

47. *Pérez-Hernández F., Tabik S., Lamas A., Olmos R., Fujita H., Herrera F.* Object detection binary classifiers methodology based on deep learning to identify small objects handled similarly: application in video surveillance // Knowl. Base Syst. 2020. N 194 P. 105590.

48. *Frickenstein A., Vemparala M.-R., Mayr J., Nagaraja N.-S., Unger C., TombariF. and StecheleW.* Binary DAD-Net: Binarized driveable area detection network for autonomous driving // IEEE Intern. Conf. Robot. Autom (ICRA). 2020. P. 2295—2301.

49. *Ajay B. S., MRao.* Binary neural network based real time emotion detection on an edge computing device to detect passenger anomaly // 34th Intern. Conf. VLSI Design, 20th Intern. Conf. Embedded Syst. (VLSID). 2021. P. 175—180.

50. *Zhuang B., Shen C., Tan M., Liu L. and Reid I.* Structured binary neural networks for accurate image classification and semantic segmentation // IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR). 2019. P. 413—422.

Сведения об авторе

Али Шаккуф — аспирант; Университет ИТМО, факультет систем управления и робототехники; E-mail: ashakkuf@itmo.ru

Поступила в редакцию 05.06.2023; одобрена после рецензирования 22.06.2023; принята к публикации 27.09.2023.