UDC 519.226+004.852
DOI: 10.20535/1810-0546.2017.1.88156

V.V. Romanuke*

Khmelnitskiy National University, Khmelnitskiy, Ukraine

APPROPRIATE NUMBER AND ALLOCATION OF ReLUs IN CONVOLUTIONAL NEURAL NETWORKS

Background. Due to that there is no common conception about whether each convolutional layer must be followed with a ReLU, the question on an appropriate number of ReLUs and their allocation is considered.
Objective. The goal is to find a law for ascertaining an appropriate number of ReLUs. If this number is less than the number of convolutional layers, then the law shall stand for an appropriate allocation of ReLUs.
Methods. A method of evaluating performance on the EEACL26 and CIFAR-10 datasets over various versions of ReLUs' allocation is defined. The performance is evaluated through 4 and 8 epochs for EEACL26 and CIFAR-10, respectively, for each allocation version. The best scores of performance are extracted.
Results. In convolutional neural networks with 4 or 5 convolutional layers, the first three convolutional layers shall be followed with ReLUs, and the rest of convolutional layers shall not be ReLUed. It is plausible that appropriateness of ReLUs includes from-the-start compactness of allocating them, i. e. all ReLUs are allocated one by one from the very first convolutional layer. An appropriate number of ReLUs is an integer between a half of the convolutional layers' number and the half increased by 1.
Conclusions. In some cases, the gain can grow up to 100 % and more. The gain for CIFAR-10, if any, is of roughly 10 to 20 %. Generally, as the training process goes on, the gain expectedly drops. Nevertheless, the stated appropriateness of number and allocation of ReLUs rationalizes the convolutional neural network architecture. Convolutional neural networks under the appropriate ReLUs' allocation can be progressively optimized further on its other hyperparameters.

Keywords: convolutional neural network; ReLU; EEACL26; CIFAR-10.

Introduction

In convolutional neural networks (CNNs), a rectified linear unit (ReLU) is a layer that applies the non-saturating activation function $f(x) = \max\{0, x\}$. It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the preceding convolution layer (ConvL). Some other functions are also used to increase nonlinearity. They are, for example, the saturating hyperbolic tangent and the sigmoid function [1, 2]. Compared to other functions the usage of ReLU is preferable, because CNNs with ReLUs are trained faster [3] without making a significant difference to generalization accuracy.

Problem statement

If a ReLU is in a CNN, it follows a ConvL. The last ConvL which actually is the fully-connected layer (FCL) is usually not followed with a ReLU [4]. However, there is no common conception about whether each ConvL, excepting FCL, must be followed with a ReLU. Some famous CNN architectures do not have any ReLUs. An example is VGGNet from Karen Simonyan and Andrew Zisserman [5] which was the runner-up in ILSVRC 2014 [6]. VGGNet is a 21-layer CNN consisting of 16 ConvLs and 5 pooling layers, and having nonetheless good performance on 1000 image categories [5, 6]. Therefore, the question on an appropriate number of ReLUs is still unanswered.

To answer the question, a law for ascertaining an appropriate number of ReLUs must be found. If this number is less than the number of ConvLs, then the law shall stand for an appropriate allocation of ReLUs. In other words, which ConvLs should not be followed with ReLUs is going to be substantiated. For achieving this, the five tasks are going to be done:

1. To define benchmark image recognition problems (IRPs) which will be exploited for gathering statistics of performance over various versions of ReLUs' allocation. Each IRP should be solved with a fixed CNN architecture, implying a constant number of ConvLs. A ReLU can be inserted wherever is needed.

2. To define a method of evaluating performance over various versions of ReLUs' allocation. The performance will be evaluated through a few epochs for each allocation version. The number of epochs is assigned for the evaluation consistency, rather than a perfect improved performance.

*corresponding author: romanukevadimv@mail.ru

3. To extract the best scores of performance and suggest a factor or a group of factors unifying those scores, related to ReLUs' allocations. This is about a common feature in those ReLUs' allocations bringing the close-to-best performance.

4. To formulate a routine of allocating ReLUs for scoring the best performance. This routine is the law prescribing positions in the CNN architecture where ReLUs should be inserted.

5. To validate the routine. The validation refers to the benchmark IRPs solved under the appropriate ReLUs' allocation against other allocations.

Benchmark IRPs

An IRP dataset should not be big for gathering statistics faster. On the other hand, the dataset performance should not be very high because we must see its significant improvement under the appropriate ReLUs' allocation. The MNIST dataset cannot be exploited due to that reason as CNNs recognize the MNIST dataset handwritten digits at 99.73 % accuracy.

Instead of MNIST, a dataset of enlarged English alphabet capital letters (EEACL26) will be used [7, 8]. Although EEACL26 is a dataset of artificial images (Fig. 1), they are featured with horizontal and vertical lines, squares, circles, crossings, diagonals, curves, serpentine lines, i. e. with all important basic attributes of real images. This dataset constituting 26 categories is simultaneously simple and useful for gathering statistics of performance in the fastest way. The EEACL26 dataset is scalable, i. e. we can generate as many images as needed.

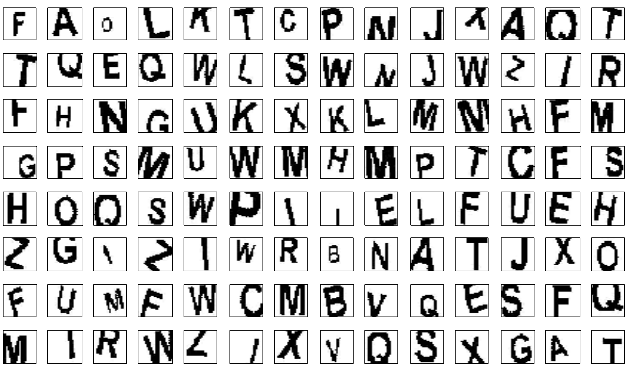

Fig. 1. A subset of 112 monochrome images of size 28×28 from the EEACL26 dataset

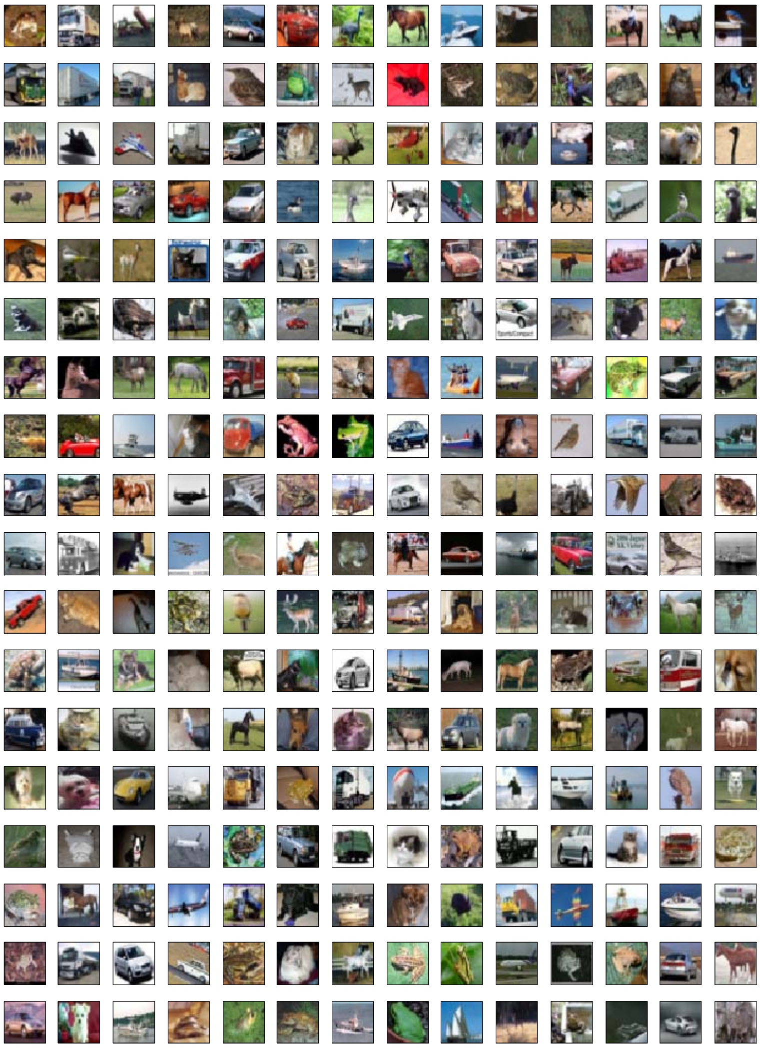

Fig. 2. Diversity of color images from the CIFAR-10 dataset containing 10 image categories (labeled as "airplane", "automobile", "bird", "cat", "deer", "dog", "frog", "horse", "ship", "truck")

Another IRP will be based on the CIFAR-10 dataset of 32×32 color images [3, 9, 10]. Its performance is far poorer than that of the EEACL26 dataset, because CIFAR-10 images are motley and heterogeneous (Fig. 2). Nevertheless, the CIFAR-10 dataset performance is expected to be more "sensitive" to the appropriate ReLUs' allocation and allocations close to the appropriateness.

A CNN for EEACL26 can be successfully trained with 4 ConvLs. CIFAR-10 needs 5 ConvLs. Note that successfulness here does not mean entirely applicable accuracy in the end of training. It means just that the training process progresses, and it is unnecessary to come to its completion.

In CNNs for the EEACL26 dataset, filters of ConvLs have size of 3 to 8, regarding that two 2×2 max pooling layers are inserted after the first and second ConvLs. For the CIFAR-10 dataset, three 2×2 max pooling layers are inserted after the first three ConvLs. Thus, here filters of ConvLs have size of 2 to 5, although size of 1 is once set in the last ConvL for 32×32 images. Numbers of filters are 20, 50, 100 for EEACL26 (it is 26 filters in the last ConvL). These numbers are 20, 50, 100, 200 for CIFAR-10 with 10 filters in the last ConvL. The training parameters are constant: the learning rate is 0.001, momentum is 0.9, and weight decay is 0.0005, which are nearly optimal [3, 5, 6, 9, 10]. Size of image batch for training is set to 100. Hyper-parameters are standard: stride is 1 for ConvLs and is 2 for pooling, without zero-padding.

Evaluation of performance

Maximal number of ReLUs equals to the number of ConvLs. Minimal number of ReLUs is 0 meaning that ReLU is not used at all. Let the k-th version of ReLUs' allocation correspond to the binary number $k-1$ represented with 4 positions for EEACL26, and with 5 positions for CIFAR-10. For instance, the 7-th version for CIFAR-10 is 00110, implying to allocate ReLUs after only the third and fourth ConvLs. The first version corresponding to 0 means a CNN without ReLU. The second version corresponding to 1 means a CNN with a single ReLU inserted after the last ConvL (FCL). The last version, which is the 16-th and 32-nd for EEACL26 and CIFAR-10, respectively, means that ReLU is inserted after each ConvL.

Performance is evaluated for each of those 16 and 32 versions. A suitable number of epochs for evaluation on EEACL26 is 4. This number is twice greater for CIFAR-10. No perfect performance is achieved through this number of epochs, but it is enough for the evaluation consistency and the corresponding inferences. However, it is not sufficient to evaluate just on such two image sizes. Therefore, initial 28×28 images from EEACL26 are resized to $N\times N$ by

$$N \in A_{\mathrm{EEACL}} = \{32, 36, 40, 44, 48, 52, 56, 60, 64\}.$$

Initial $32\times32\times3$ images from CIFAR-10 are resized to $N\times N\times3$ by

$$N \in A_{\mathrm{CIFAR}} \in \{36, 40, 44, 48, 52, 56, 60, 64\}.$$

The resulting performance is of vectors

$$\mathbf{V}_{\mathrm{EEACL}}(N, k) = [v_p^*(N, k)]_{1\times4} \tag{1}$$

and

$$\mathbf{V}_{\mathrm{CIFAR}}(N, k) = [v_p^{**}(N, k)]_{1\times8}, \tag{2}$$

where $v_p^*(N, k)$ is an error rate on EEACL26 after the p-th epoch, and $v_p^{**}(N, k)$ is an error rate on CIFAR-10 after the p-th epoch. To compare them independently of image size and dataset, they are normalized. Normalization of averages gives

$$\tilde{v}^*(N, k) = \frac{\sum_{p=1}^{4} v_p^*(N, k)}{\max\limits_{q=1,\,16} \sum_{p=1}^{4} v_p^*(N, q)} \tag{3}$$

and

$$\tilde{v}^{**}(N, k) = \frac{\sum_{p=1}^{8} v_p^{**}(N, k)}{\max\limits_{q=1,\,32} \sum_{p=1}^{8} v_p^{**}(N, q)}. \tag{4}$$

Along with (3) and (4), the normalized performance after the final epoch should be used also:

$$\tilde{v}_4^*(N, k) = \frac{v_4^*(N, k)}{\max\limits_{q=1,\,16} v_4^*(N, q)} \tag{5}$$

and

$$\tilde{v}_8^{**}(N, k) = \frac{v_8^{**}(N, k)}{\max\limits_{q=1,\,32} v_8^{**}(N, q)}. \tag{6}$$

Owing to that each of the normalized error rates (3)—(6) achieves its maximal value equal to 1, they can be summed over N. This is why evaluations

$$\tilde{v}^*(k) = \sum_{N\in\{28\}\cup A_{\mathrm{EEACL}}} \tilde{v}^*(N, k), \tag{7}$$

$$\tilde{v}^{**}(k) = \sum_{N\in\{32\}\cup A_{\mathrm{CIFAR}}} \tilde{v}^{**}(N, k), \tag{8}$$

$$\tilde{v}_4^*(k) = \sum_{N\in\{28\}\cup A_{\mathrm{EEACL}}} \tilde{v}_4^*(N, k), \tag{9}$$

$$\tilde{v}_8^{**}(k) = \sum_{N\in\{32\}\cup A_{\mathrm{CIFAR}}} \tilde{v}_8^{**}(N, k) \tag{10}$$

are eligible for finding appropriate allocations of ReLUs.

Best scores of performance

The best scores of performance are extracted by minimizing the evaluations (7)—(10). Fig. 3 and Fig. 4 show 10 polylines (3) and 9 polylines (4) plotted against versions of ReLUs' allocation, respectively. Polylines (3) are very scattered because there are six cases of fails in training on the EEACL26 dataset:

1. $N = 44$, $k = 9$ (but just after the fourth epoch, that looks like a casual fail, not systematic, although it is a failure for this k).
2. $N = 48$, $k \in \{1, 2\}$.
3. $N = 52$, $k \in \{1, 2\}$.
4. $N = 56$, $k \in \{1, 2\}$.
5. $N = 60$, $k \in \{1, 2, 11\}$.
6. $N = 64$, $k \in \{1, 2, 9, 10\}$.

These fails indicate at that CNNs without ReLUs or CNNs with a single ReLU inserted after the last ConvL cannot perform good. Allocating a single ReLU in the very start which is $k = 9$ for the allocation by the binary number $1000_{(2)}$ is poor practice. Besides, allocating ReLUs from both ends which is $k = 10$ for $1001_{(2)}$ is unacceptable. A version with an alternation like at $k = 11$ for $1010_{(2)}$ does not seem promising. The same concerns $k = 12$ for $1011_{(2)}$ and $k = 16$ for $1111_{(2)}$.

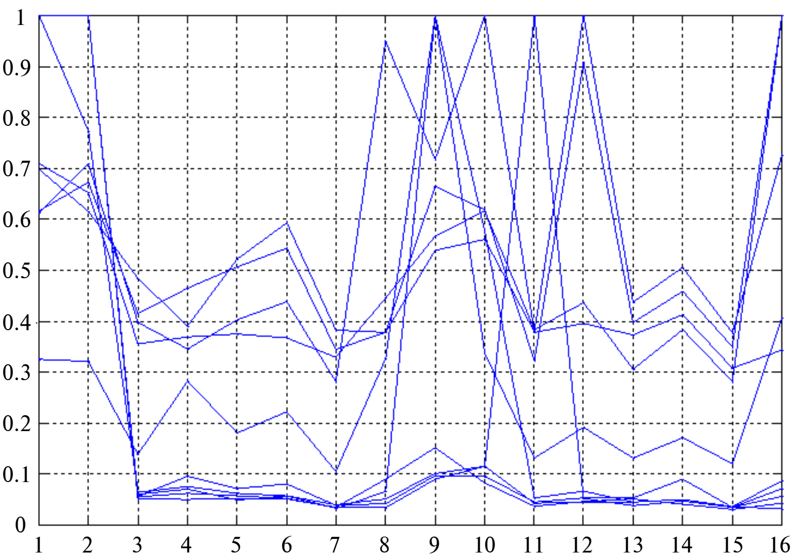

Fig. 3. The 10 polylines (3) plotted against 16 versions of ReLUs' allocation

Unlike the EEACL26 dataset, the CIFAR-10 dataset gives more controversial results for $k = 1$, when CNN does not have ReLUs. Although performance on CIFAR-10 is not so scattered, it is harder to infer from polylines in Fig. 4. However, the most protuberant points (whose bunch is closer to 1) can be marked out for

$$k \in \{2, 4, 12, 16, 20, 24, 28, 32\}$$

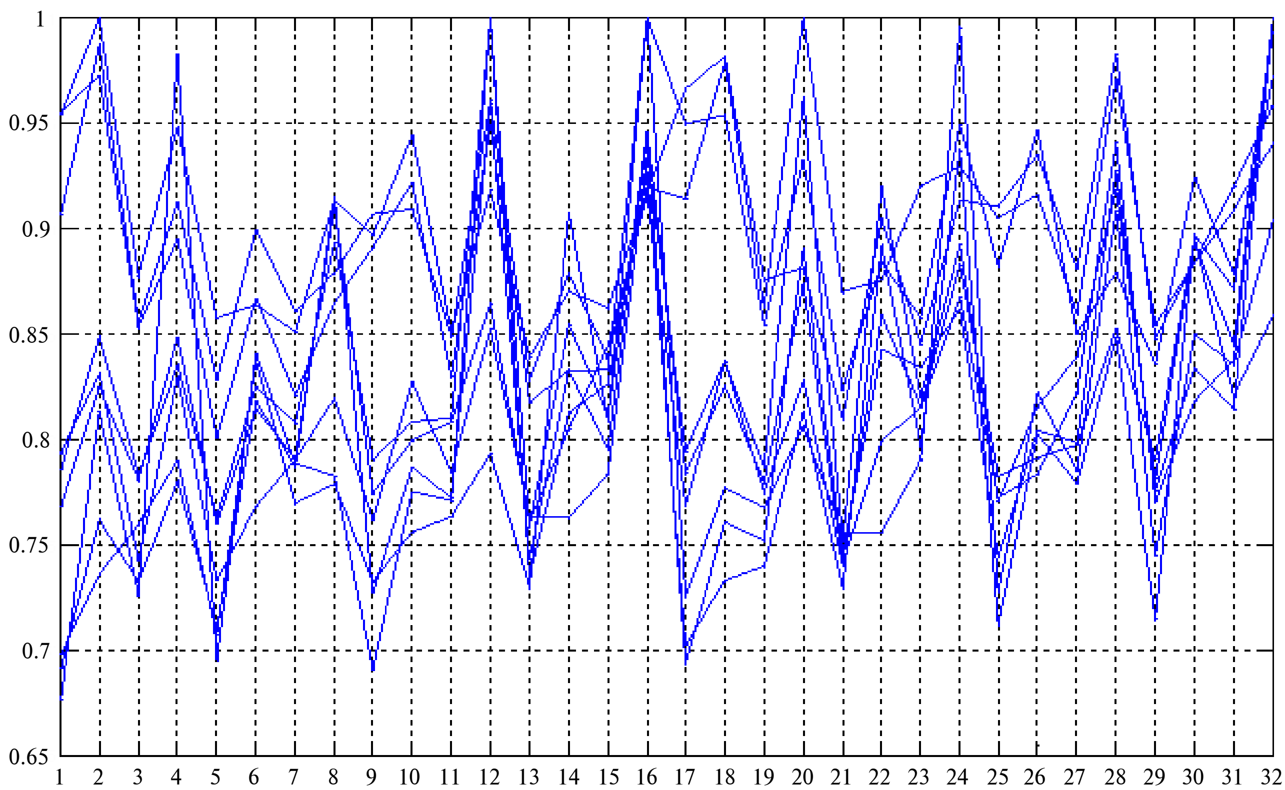

Fig. 4. The 9 polylines (4) plotted against 32 versions of ReLUs' allocation

by the ReLUs' allocations corresponding to

$$00001_{(2)},\ 00011_{(2)},\ 01011_{(2)},\ 01111_{(2)}, 10011_{(2)},\ 10111_{(2)},\ 11011_{(2)},\ 11111_{(2)}\,. \qquad (11)$$

As it is seen, the worst scores for the CIFAR-10 dataset have the factor of the last ConvL's ReLU unifying them with the EEACL26 worst scores. Another unifying factor is coming out of $1111_{(2)}$ and $11111_{(2)}$ as the fully-ReLUed CNNs perform poorer. Moreover, allocating a couple of ReLUs in the two last ConvLs gives poor performance also, no matter how many ReLUs precede them.

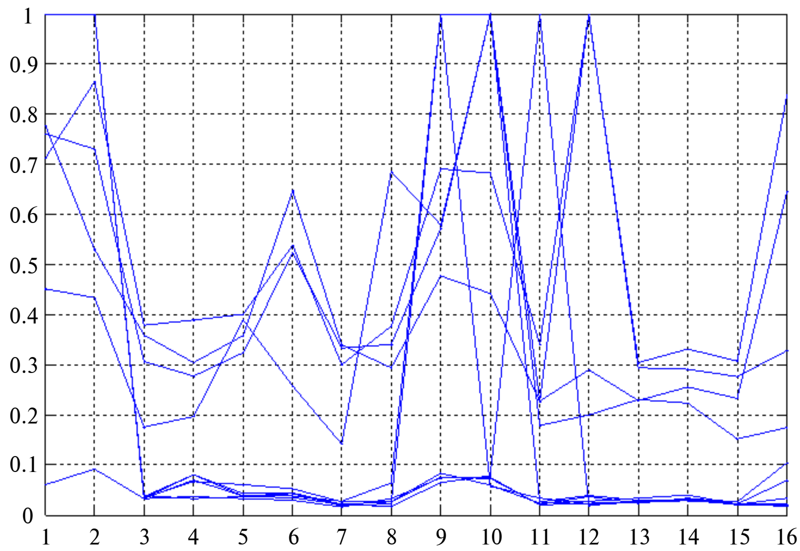

Fig. 5. The final fourth epoch performance (5) against 16 versions

Polylines (5) and (6) reflecting the final epoch performance are shown similarly in Fig. 5 and Fig. 6. These graphs are just inside of the sums (7)—(10) which are shown in Figs. 7—10. In accordance with Figure 5, the most "stable" versions of allocating ReLUs are

$$0010_{(2)},\ 0011_{(2)},\ 0100_{(2)},\ 0110_{(2)},\ 1100_{(2)},\ 1101_{(2)},\ 1110_{(2)}\,, \qquad (12)$$

that factually contradicts some versions in (11). The "stability" is hardly seen in Fig. 6, although the scattering here is lesser than that in Fig. 5.

The best performance for EEACL26 is 1.54 % by $N = 52$, and the best performance is roughly bettering as N increases. The best performance for CIFAR-10 is 24.41 % by $N = 64$, and it is bettering also as N increases, but the error rate decrement is stronger apparent.

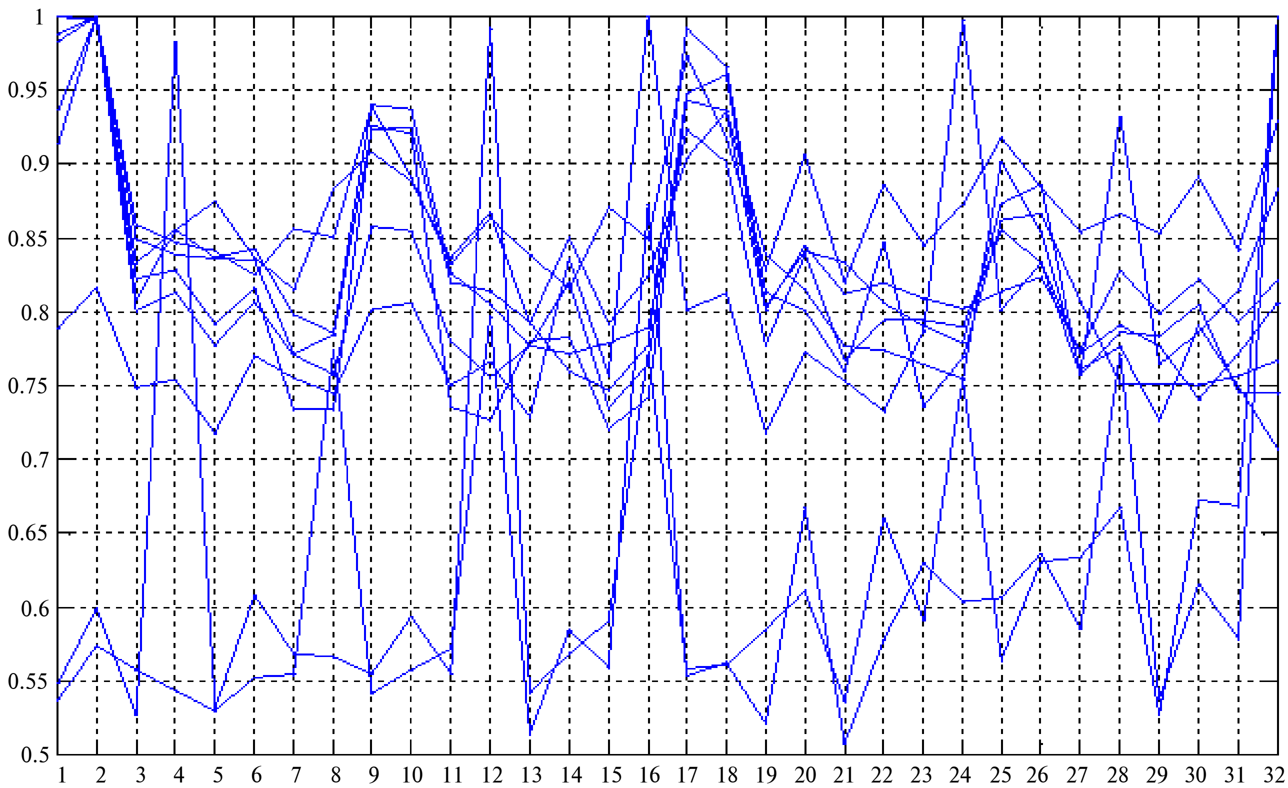

Fig. 6. The final eighth epoch performance (6) against 32 versions

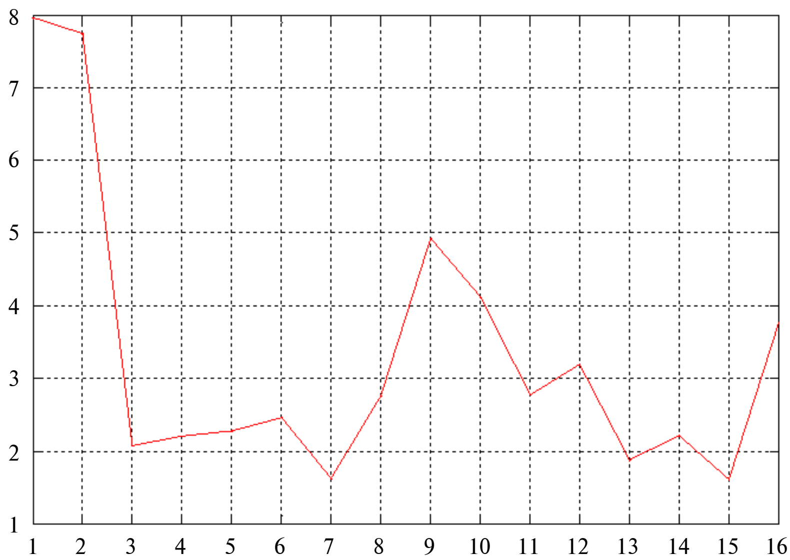

Fig. 7. The sum (7) of 10 polylines (3) in Fig. 3

The sums in Figs. 7 and 9 allow to conclude that the most appropriate versions of allocating ReLUs for EEACL26 are

$$0110_{(2)},\ 1100_{(2)},\ 1101_{(2)},\ 1110_{(2)}\,, \tag{13}$$

that is a subset of those ones in the list (12). Non-appropriateness of allocations

$$0001_{(2)},\ 0101_{(2)},\ 1000_{(2)},\ 1001_{(2)},\ 1010_{(2)},\ 1011_{(2)},\ 1111_{(2)}\,, \tag{14}$$

and of non-allocation $0000_{(2)}$ is confirmed.

Some of the extracted best scores in Figs. 8 and 10 are contradictious. They are

$$00100_{(2)},\ 00110_{(2)},\ 01110_{(2)}\,. \tag{15}$$

This is why allocations (15) cannot be considered for being appropriate. The most appropriate versions of allocating ReLUs for CIFAR-10 are

$$01100_{(2)},\ 10110_{(2)},\ 11100_{(2)}\,. \tag{16}$$

Non-appropriateness of allocations (11) along with

$$00111_{(2)},\ 01000_{(2)},\ 01001_{(2)},\ 10000_{(2)},\ 10001_{(2)},\ 11000_{(2)},\ 11001_{(2)}\,, \tag{17}$$

and the non-allocation $00000_{(2)}$ is confirmed as well. Allocations $11101_{(2)}$ and $11110_{(2)}$ which are closer to the fully-ReLUed CNNs are unreasonable.

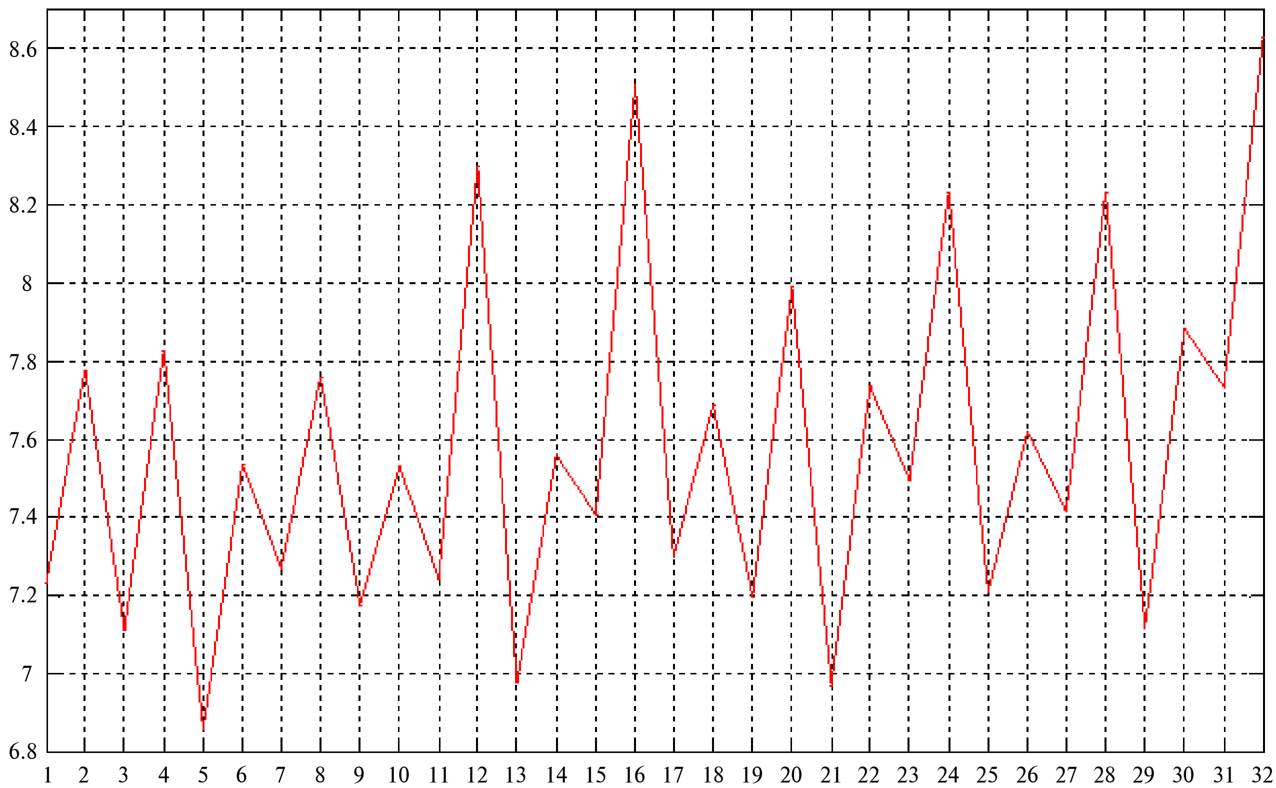

Fig. 8. The sum (8) of 9 polylines (4) in Fig. 4

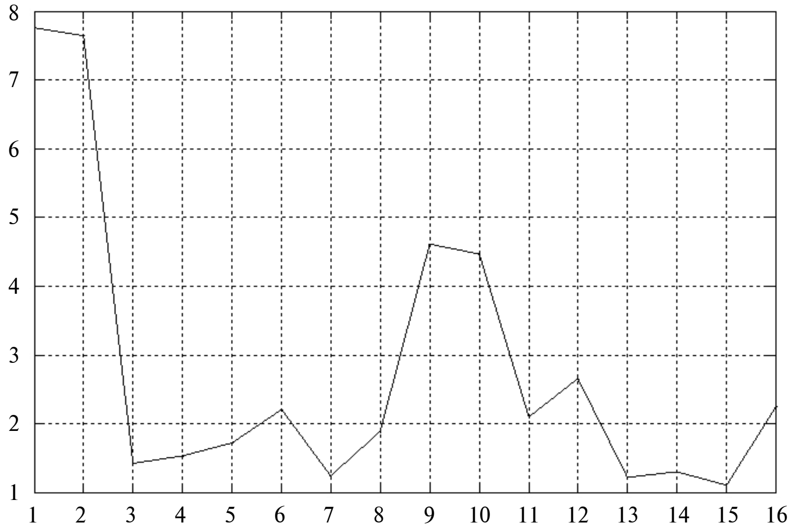

Fig. 9. The sum (9) of 10 polylines (5) in Fig. 5

Appropriateness of ReLUs

Now the task is to see factors which are common for (13) and (16). But also non-appropriateness in (14) and (17) should be regarded. Thus, the case of only two ReLUs in the very start can be treated as a non-appropriate allocation due to allocations $1100_{(2)}$ and $11000_{(2)}$ are contradictious. Allocating ReLUs in the middle like $0110_{(2)}$ and $01110_{(2)}$ does not seem promising. Further, version $1101_{(2)}$ is nothing but a single allocation among those ones in (13) and (16) which proposes to insert a ReLU after the last ConvL.

As allocations $1110_{(2)}$ and $11100_{(2)}$ bring the close-to-best performance, then the first three ConvLs shall be followed with ReLUs, and the rest of ConvLs shall not be ReLUed. This is an appropriate number and allocation of ReLUs in CNNs with 4 or 5 ConvLs. It is plausible that appropriateness of ReLUs in CNNs with more ConvLs includes from-the-start compactness of allocating ReLUs, i. e. all ReLUs are allocated one by one from the very first ConvL. An appropriate number of ReLUs is an integer belonging to the segment

$$\left[\frac{L_{\text{Conv}}}{2};\ \frac{L_{\text{Conv}}}{2}+1\right] \tag{18}$$

by L_{Conv} ConvLs, where $L_{\text{Conv}} \in \mathbb{N} \setminus \{1\}$. When number L_{Conv} is even, there are two versions of ReLUs' number. That actually was revealed for EEACL26 with $1100_{(2)}$ and $1110_{(2)}$. Therefore, CNNs consisting of 2 ConvLs should have a single ReLU or be the fully-ReLUed CNNs.

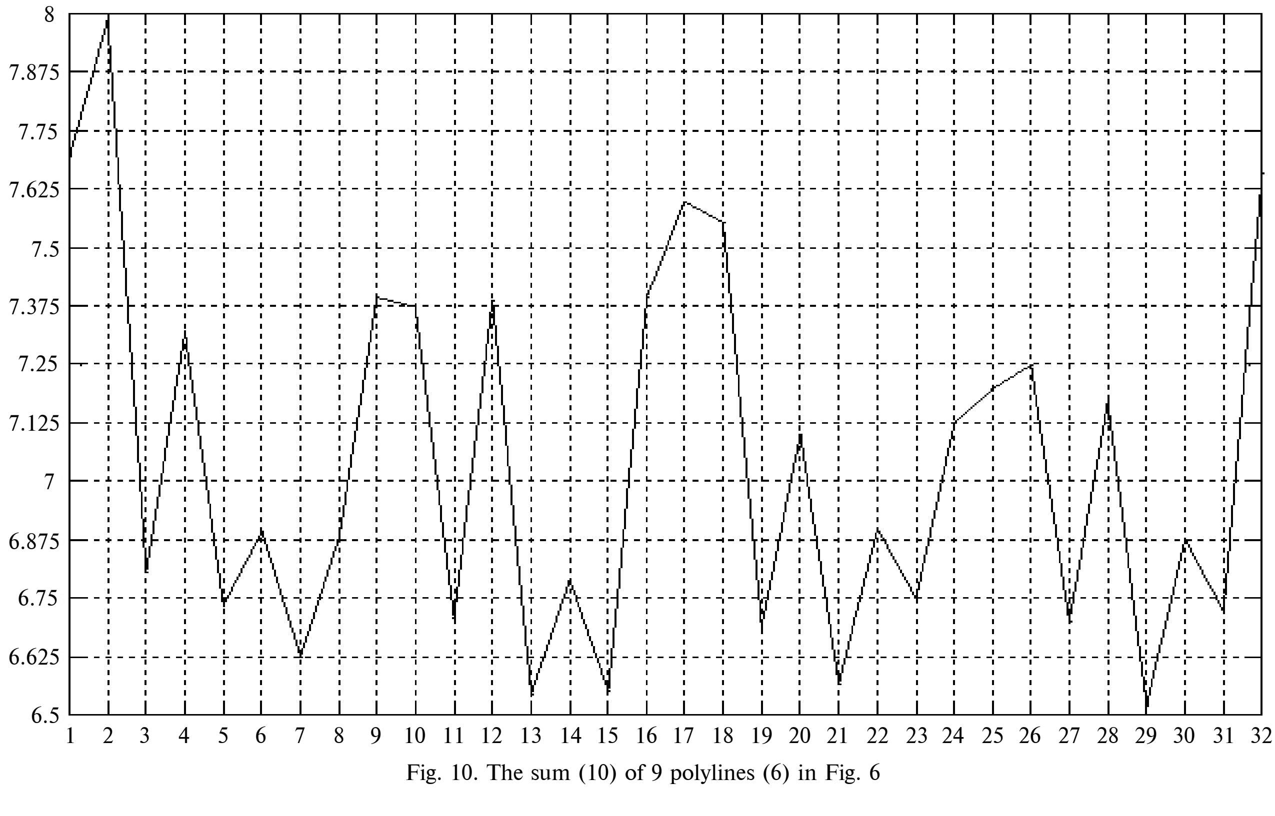

Fig. 10. The sum (10) of 9 polylines (6) in Fig. 6

Validation

The formulated routine of allocating ReLUs for scoring the best performance is validated on the benchmark IRPs of EEACL26 and CIFAR-10 datasets by training in 24 epochs. The relative difference between performance by the appropriate ReLUs' allocation and a non-appropriate ReLUs' allocation is shown against epochs in Figs. 11 and 12, where points above the horizontal line imply positive gains standing for the formulated routine.

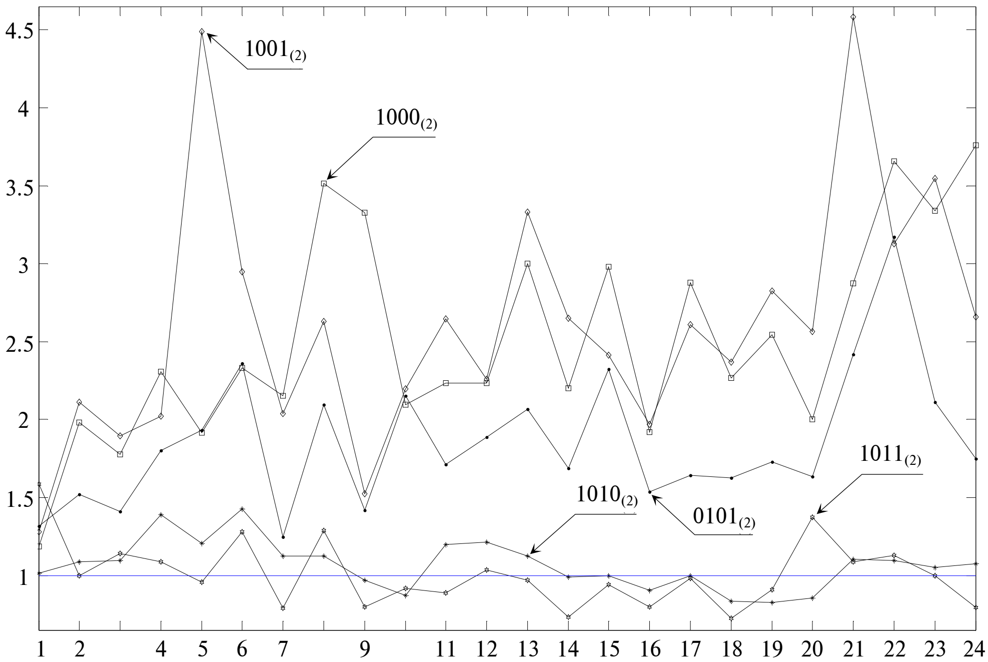

Fig. 11. Ratios of the normalized epoch-wise performance for the EEACL26 dataset by non-appropriate ReLUs' allocations to the normalized epoch-wise performance by the appropriate ReLUs' allocation

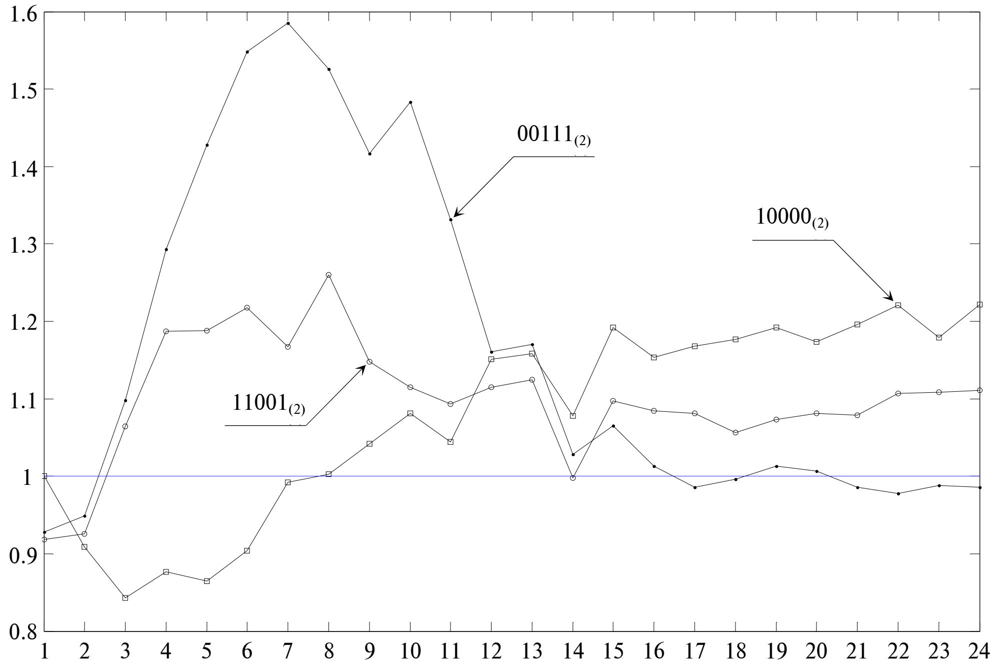

Fig. 12. Ratios of the normalized epoch-wise performance for the CIFAR-10 dataset by non-appropriate ReLUs' allocations to the normalized epoch-wise performance by the appropriate ReLUs' allocation

Except the allocation $10000_{(2)}$, the positive gain for CIFAR-10 is apparent right after the epochs 3—13. Since the 14-th epoch, the gain is not so obvious, but it still is convincing. Unlike allocations $00111_{(2)}$ and $11001_{(2)}$, non-appropriateness of the allocation $10000_{(2)}$ reveals itself later. The gain for EEACL26 is far stronger, except the far weaker non-appropriateness of the allocations $1010_{(2)}$ and $1011_{(2)}$. Generally, as the training process goes on, the gain expectedly drops. Eventually, the appropriateness is not optimality, so some IRPs may be ReLUed in a different way.

Conclusions

For significant improvement of performance, ReLUs should be inserted one by one from the very first ConvL to cover approximately a half of ConvLs. The appropriate number of ReLUs is either the middle of the segment (18) for odd L_{Conv} or both the ends of this segment for even L_{Conv}. Fig. 11 has shown that, in some cases, the gain can grow up to 100 % and more. The gain for CIFAR-10, if any (see Fig. 12), is of roughly 10 % to 20 %. But the gain may fade away if the training is too long. Nevertheless, the stated appropriateness of number and allocation of ReLUs rationalizes the CNN architecture. CNNs under the appropriate ReLUs' allocation can be progressively optimized further on its other hyperparameters.

List of literature

1. *Costarelli D., Spigler R.* Approximation results for neural network operators activated by sigmoidal functions // Neural Networks. – 2013. – **44**. – P. 101—106. doi: 10.1016/j.neunet.2013.03.015
2. *Anastassiou G.A.* Multivariate sigmoidal neural network approximation // Neural Networks. – 2011. – **24**, iss. 4. – P. 378—386. doi: 10.1016/j.neunet.2011.01.003
3. *Krizhevsky A., Sutskever I., Hinton G.* ImageNet classification with deep convolutional neural networks // Advances in Neural Information Processing Systems. – 2012. – **1**. – P. 1097—1105.
4. *Convolutional* neural network with biologically inspired retinal structure/ J. Kim, S. O, Y. Kim, M. Lee // Procedia Comp. Sci. – 2016. – **88**. – P. 145—154. doi: 10.1016/j.procs.2016.07.418
5. *Simonyan K., Zisserman A.* Very deep convolutional networks for large-scale image recognition // 5th Int. Conf. Learning Representations (ICLR 2015). – 2015. – arXiv:1409.4842.
6. *Going* deeper with convolutions / C. Szegedy, W. Liu, Y. Jia et al. // 2015 IEEE Conf. Computer Vision and Pattern Recognition (CVPR). – 2015. – arXiv:1409.4842. doi: 10.1109/CVPR.2015.7298594
7. *Romanuke V.V.* Two-layer perceptron for classifying flat scaled-turned-shifted objects by additional feature distortions in training // J. Uncertain Systems. – 2015. – **9**, № 4. – P. 286—305.
8. *Romanuke V.V.* Optimal training parameters and hidden layer neurons number of two-layer perceptron for generalized scaled objects classification problem // Inform. Technol. Management Sci. – 2015. – **18**. – P. 42—48.
9. *Flexible*, high performance convolutional neural networks for image classification / D. Ciresan, U. Meier, J. Masci et al. // Proc. XXII Int. Joint Conf. Artificial Intelligence. – 2011. – **2**. – P. 1237—1242.
10. *Date P., Hendler J.A., Carothers C.D.* Design index for deep neural networks // Procedia Comp. Sci. – 2016. – **88**. – P. 131—138. doi: 10.1016/j.procs.2016.07.416

References

[1] D. Costarelli and R. Spigler, "Approximation results for neural network operators activated by sigmoidal functions", *Neural Networks*, vol. 44, pp. 101—106, 2013. doi: 10.1016/j.neunet.2013.03.015

[2] G.A. Anastassiou, "Multivariate sigmoidal neural network approximation", *Neural Networks*, vol. 24, iss. 4, pp. 378—386, 2011. doi: 10.1016/j.neunet.2011.01.003

[3] A. Krizhevsky *et al.*, "ImageNet classification with deep convolutional neural networks", *Advances in Neural Information Processing Systems*, vol. 1, pp. 1097—1105, 2012.

[4] J. Kim *et al.*, "Convolutional neural network with biologically inspired retinal structure", *Procedia Comp. Sci.*, vol. 88, pp. 145—154, 2016. doi: 10.1016/j.procs.2016.07.418

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", in *Proc. 5th Int. Conf. Learning Representations (ICLR 2015)*, arXiv:1409.1556v6, 2015.

[6] C. Szegedy *et al.*, “Going deeper with convolutions”, in *Proc. 2015 IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, arXiv:1409.4842, 2015. doi: 10.1109/CVPR.2015.7298594

[7] V.V. Romanuke, “Two-layer perceptron for classifying flat scaled-turned-shifted objects by additional feature distortions in training”, *J. Uncertain Systems*, vol. 9, no. 4, pp. 286−305, 2015.

[8] V.V. Romanuke, “Optimal training parameters and hidden layer neurons number of two-layer perceptron for generalized scaled objects classification problem”, *Inform. Technol. Management Sci.*, vol. 18, pp. 42−48, 2015.

[9] D. Ciresan *et al.*, “Flexible, high performance convolutional neural networks for image classification”, in *Proc. XXII Int. Joint Conf. Artificial Intelligence*, vol. 2, pp. 1237−1242, 2011.

[10] P. Date *et al.*, “Design index for deep neural networks”, *Procedia Comp. Sci.*, vol. 88, pp. 131−138, 2016. doi: 10.1016/j.procs.2016.07.416

В.В. Романюк

НАЛЕЖНЕ ЧИСЛО І РОЗМІЩЕННЯ ВУЗЛІВ ЛІНІЙНИХ ВИПРЯМЛЯЧІВ У ЗГОРТКОВИХ НЕЙРОННИХ МЕРЕЖАХ

Проблематика. Оскільки не існує загальної концепції того, чи повинен кожен згортковий шар супроводжуватися вузлом лінійного випрямляча, розглядається питання щодо належного числа вузлів лінійних випрямлячів та їх розміщення.

Мета дослідження. Метою роботи є знаходження закону для встановлення належного числа вузлів лінійних випрямлячів. Якщо це число є меншим за число згорткових шарів, то цей закон обумовлюватиме належне розміщення вузлів лінійних випрямлячів.

Методика реалізації. Визначається метод оцінювання продуктивності на банках даних EEACL26 та CIFAR-10 за різними версіями розміщення вузлів лінійних випрямлячів. Для кожної версії розміщення продуктивність оцінюється упродовж 4 та 8 епох відповідно для EEACL26 та CIFAR-10. Виокремлюються найкращі показники продуктивності.

Результати дослідження. У згорткових нейронних мережах з 4 та 5 згортковими шарами перші три згорткові шари мають супроводжуватися вузлами лінійних випрямлячів, а решта згорткових шарів не повинна піддаватися цій обчислювальній процедурі. Правдоподібно те, що доцільність вузлів лінійних випрямлячів включає їх компактне розміщення з початку, тобто всі вузли лінійних випрямлячів розміщуються один за одним від самого першого згорткового шару. Належне число вузлів лінійних випрямлячів є деяким цілим числом між половиною кількості згорткових шарів та цією половиною, збільшеною на 1.

Висновки. У деяких випадках виграш може доходити до 100 % і більше. Виграш для CIFAR-10, якщо такий матиме місце, становить від близько 10 до 20 %. Взагалі, виграш очікувано спадає з просуванням процесу навчання. Однак наведена доцільність числа та розміщення вузлів лінійних випрямлячів раціоналізує архітектуру згорткових нейронних мереж. За належного розміщення вузлів лінійних випрямлячів згорткові нейронні мережі можуть бути поступово оптимізовані далі за іншими своїми гіперпараметрами.

Ключові слова: згорткова нейронна мережа; ReLU; EEACL26; CIFAR-10.

В.В. Романюк

НАДЛЕЖАЩЕЕ ЧИСЛО И РАСПОЛОЖЕНИЕ УЗЛОВ ЛИНЕЙНЫХ ВЫПРЯМИТЕЛЕЙ В СВЕРТОЧНЫХ НЕЙРОННЫХ СЕТЯХ

Проблематика. Поскольку не существует общей концепции того, должен ли каждый сверточный шар сопровождаться узлом линейного выпрямителя, рассматривается вопрос касательно надлежащего числа узлов линейных выпрямителей и их расположения.

Цель исследования. Целью работы является нахождение закона для определения надлежащего числа узлов линейных выпрямителей. Если это число меньше, чем число сверточных шаров, то этот закон будет обуславливать надлежащее расположение узлов линейных выпрямителей.

Методика реализации. Определяется метод оценивания производительности на наборах данных EEACL26 и CIFAR-10 по различным версиям расположения узлов линейных выпрямителей. Для каждой версии расположения производительность оценивается в течение 4 и 8 эпох соответственно для EEACL26 и CIFAR-10. Извлекаются наилучшие показатели производительности.

Результаты исследования. В сверточных нейронных сетях с 4 и 5 сверточными шарами первые три сверточных шара должны сопровождаться узлами линейных выпрямителей, а остальные сверточные шары не поддаются этой вычислительной процедуре. Правдоподобно то, что целесообразность узлов линейных выпрямителей включает их компактное расположение с начала, то есть все узлы линейных выпрямителей размещаются один за другим с самого первого сверточного шара. Надлежащее число узлов линейных выпрямителей является некоторым целым числом между половиной количества сверточных шаров и этой половиной, увеличенной на 1.

Выводы. В некоторых случаях выигрыш может доходить до 100 % и больше. Выигрыш для CIFAR-10, если таковой будет, составляет от приблизительно 10 до 20 %. Вообще, выигрыш ожидаемо падает при продвижении процесса обучения. Тем не менее изложенная целесообразность числа и расположения узлов линейных выпрямителей рационализирует архитектуру сверточных нейронных сетей. При надлежащем расположении узлов линейных выпрямителей сверточные нейронные сети могут быть постепенно оптимизированы по другим своим гиперпараметрам.

Ключевые слова: сверточная нейронная сеть; ReLU; EEACL26; CIFAR-10.

Рекомендована Радою
факультету прикладної математики
КПІ ім. Ігоря Сікорського

Надійшла до редакції
28 грудня 2016 року