sciendo RIGA TECHNICAL UNIVERSITY *Electrical, Control and Communication Engineering*

ISSN 2255-9159 (online)
ISSN 2255-9140 (print)
2018, vol. 14, no. 1, pp. 51–57
doi: 10.2478/ecce-2018-0006
https://www.degruyter.com/view/j/ecce

An Attempt of Finding an Appropriate Number of Convolutional Layers in CNNs Based on Benchmarks of Heterogeneous Datasets

Vadim V. Romanuke* (*Professor, Polish Naval Academy, Gdynia, Poland*)

Abstract **– An attempt of finding an appropriate number of convolutional layers in convolutional neural networks is made. The benchmark datasets are CIFAR-10, NORB and EEACL26, whose diversity and heterogeneousness must serve for a general applicability of a rule presumed to yield that number. The rule is drawn from the best performances of convolutional neural networks built with 2 to 12 convolutional layers. It is not an exact best number of convolutional layers but the result of a short process of trying a few versions of such numbers. For small images (like those in CIFAR-10), the initial number is 4. For datasets that have a few tens of image categories and more, initially setting five to eight convolutional layers is recommended depending on the complexity of the dataset. The fuzziness in the rule is not removable because of the required diversity and heterogeneousness.**

Keywords **– Convolutional neural networks; Convolutional layers; Error rate; Hyperparameters; Performance.**

I. The Problem of an Appropriate Number of Convolutional Layers

In machine learning for image recognition, the convolutional layer (ConvL) is the core building block of a convolutional neural network (CNN). A ConvL is a set of learnable filters which actually are three-dimensional matrices, to which a bias vector is attached [1], [2]. The parameters of a ConvL, called hyperparameters, are as follows [2], [3]:

1. Height F_{height} of the filter (size along the vertical axis). Integer F_{height} must be positive.
2. Width F_{width} of the filter (the horizontal axis). Integer F_{width} must be positive, and commonly $F_{\text{width}} = F_{\text{height}}$ [4], [5].
3. Depth K_{ConvL} of the filter. The depth of the filter of the first ConvL is equal to the number of colour channels in the input image. The depth of the filter of a subsequent ConvL is equal to the number of filters of the antecedent ConvL [6].
4. Stride s_{ConvL}. Integer s_{ConvL} must be positive for controlling how depth columns are allocated around the spatial dimensions (width and height). Often $s_{\text{ConvL}} = 1$, so then a new depth column of neurons is allocated to spatial positions only one spatial unit apart [7].
5. Zero-padding p_{ConvL}. Integer p_{ConvL} must be non-negative for preserving exactly the spatial size of the output volumes [2], [5], [8].

All these hyperparameters are set by rules of thumb [2], [7]. Moreover, when CNN architecture is built, the number of ConvLs N_{ConvL} (a positive integer) is set just by experience. Thus, setting the integer N_{ConvL} appropriately is an open issue. Answering this question can significantly improve performance.

II. Background and Motivation

It is believed that complexity of an image recognition problem (IRP) is associated with the number of ConvLs. The complexity of IRPs issues from the number of image categories, the number of features (dimensionality), the influence of colour, the influence of chrominance, diversities in images labelled as belonging to the same category [9], [10]. The more complex IRPs may naïvely need a greater N_{ConvL}. This has, however, not been proved yet. Moreover, it is unknown whether this is provable or not [11].

Unlike its hyperparameters, the number of ConvLs is not limited from above [1], [2], [6], [7]. If the hyperparameters are selected appropriately, N_{ConvL} should be varied starting from 2 up to some integer $N_{\text{ConvL}}^{\langle\max\rangle}$, at which the effectiveness of CNNs is less than at $N_{\text{ConvL}}^{\langle\max\rangle} - 1$. The effectiveness means performance and operation speed (computational rate) [1], [2], [5], [10], [12], [13]. Obviously, the computational rate slightly (at least) decreases as N_{ConvL} increases, so this is a constraint preventing the assigning of a great N_{ConvL} [6], [7], [9]. For instance, the position of the runner-up in ILSVRC 2014 was taken by the CNN that became known as VGGNet [11], [14] containing 16 ConvLs. A downside of VGGNet is that it is very expensive to evaluate and uses much more memory and parameters (a MATLAB .mat file of VGGNet has the size of about 1 GB). But if some ConvLs nearest to the VGGNet output layer are removed, the performance is still the same and the number of necessary parameters is significantly reduced [11], [15], [16].

III. A Goal for Finding a Rule of Appropriately Setting the Number of ConvLs

The goal is to find a rule for appropriately setting the integer N_{ConvL} regarding the number of image categories and the dimensionality of an IRP. In other words, once an IRP is given with its number of image categories and image size, the rule must yield a certain integer N_{ConvL} or a few versions of this

* E-mail: romanukevadimv@gmail.com

number. In the worst case, an integer interval for an appropriate number of ConvLs should be formed.

For stating the rule, four tasks need to be accomplished.

1. To form a variety of IRPs for benchmarking.
2. To test the IRPs on an admissible interval of integers N_{ConvL}.
3. To establish the correspondence of the best performance to N_{ConvL}.
4. To formalise the correspondence as a rule.

The rule will allow rationally constructing a pivot of CNNs which is a sequence of ConvLs. Having the pivot, the remaining parts of the CNNs (pooling layers, ReLUs, DropOut layers, normalisation layers) are allocated easier. This would be a profound contribution to the theory of CNNs for making image recognition more effective.

IV. IRPs for Benchmarking

The rule is expected to be generally acceptable for a wide range of IRPs. That is, it must be generalisable. To prevent an IRP from overfitting (this is a meta-overfitting to a group of IRPs – an extension of the common overfitting to training sets), the benchmark IRPs should be dissimilar. Thus, the IRP datasets with their entries should satisfy a requirement of dissimilarity in the following:

1) the number of image categories;
2) the number of colour channels;
3) the initial image size;
4) the origination of the image content;
5) the types of objects to be recognised.

These five dissimilarities ensure diversity and heterogeneousness to IRPs. However, this is not sufficient for benchmarking, since, for instance, the ImageNet dataset is too huge for statistical research. Therefore, an additional requirement is that the size of the benchmark IRP should be moderate. This implies a medium image size (not larger than 128 pixels) as well as a fairly small number of image categories (a few tens at the most).

There are three datasets that completely satisfy these requirements: CIFAR-10 (Fig. 1), NORB (Fig. 2), EEACL26 (Fig. 3). Although CIFAR-10 has only 10 image categories, the diversity of its entries is the highest. The CIFAR-10 image categories labelled as “airplane”, “automobile”, “bird”, “cat”, “deer”, “dog”, “frog”, “horse”, “ship”, “truck” are diverse themselves. CIFAR-10 consists of 60 000 images, where each category is represented with 6000 entries.

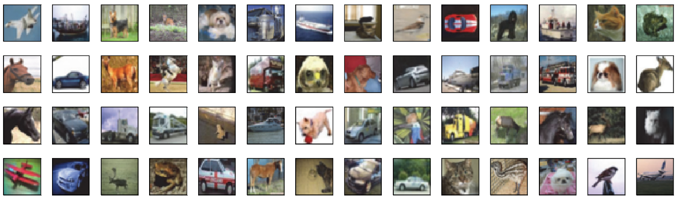

Fig. 1. A subset of the CIFAR-10 dataset consisting of colour images whose original size is 32×32 in each of the three colour channels [6], [9], [10]. The diversity of its entries is highest as the CIFAR-10 dataset is heterogeneous itself.

The NORB dataset consists of 349 920 images (with a total of 291 600 images served for training) representing fifty toys belonging to five generic categories (four-legged animals, human figures, airplanes, trucks and cars). Although NORB has only six image categories included one image background category, the diversity of its entries is rather high. The NORB objects were originally imaged by two cameras at six sets of lighting conditions, nine elevations, and eighteen azimuths. Then they were jittered and cluttered by random perturbation of position, scaling, varying brightness and contrast. The disparities were adjusted and randomly picked so that the objects appeared placed on highly textured horizontal surfaces at a small random distance from those surfaces. In addition, a randomly picked distracting object was placed at the periphery of the image.

Fig. 2. A subset of the NORB dataset consisting of 108×108 8-bit greyscale images [6]. The diversity here is high but NORB has only six image categories.

A far lighter and easier dataset is EEACL26, which represents images of enlarged capital letters of the English alphabet. It has 26 categories, and it is a completely artificial dataset, and hence it is scalable – as many EEACL26 images can be generated as needed, and their size is adjusted. There are three types of distortion – scaling, rotation, shifting. The intensity of these distortions is regulated with their magnitudes. Fig. 3 shows a moderate intensity of the distortions. At such intensity, 52 000 EEACL26 entries (2000 entries per letter) are enough for training and validating [13], [17], [18].

V. Admissibility of Integers N_{ConvL}

Admissibility here implies rationality and reasonability, i.e. testing the IRPs on an admissible interval of integers N_{ConvL} must expose the best performance as well as a moderate one, while the worse performance is expected closer to the endpoints of the interval. Setting a single ConvL is obviously inappropriate (there would not have been any convolution), so let $N_{\mathrm{ConvL}} = 2$ be the left endpoint of the interval for the worst-case reference. The maximum integer $N_{\mathrm{ConvL}}^{\langle \max \rangle}$ depends on the IRP and its image size. The entries of CIFAR-10 are recognised successfully by four to six ConvLs for any image size between 32×32 and 64×64. The same goes for EEACL26. For successful training on the NORB dataset, some versions of CNNs have only three ConvLs [3]. Eventually, the number of ConvLs is also adjusted with the number of pooling layers which follow the ConvLs. Hence, let $N_{\mathrm{ConvL}}^{\langle \max \rangle} = 8$ for 32×32 images by applying no resizing for CIFAR-10 and downsampling the NORB entries. Then let $N_{\mathrm{ConvL}}^{\langle \max \rangle} = 9$ for 48×48 images and $N_{\mathrm{ConvL}}^{\langle \max \rangle} = 10$ for 64×64 images by upsampling the CIFAR-10 entries and downsampling the NORB entries. It is appropriate to set $N_{\mathrm{ConvL}}^{\langle \max \rangle} = 11$ for 96×96 images. Separately, $N_{\mathrm{ConvL}}^{\langle \max \rangle} = 12$ for the original NORB 108×108 images. All the versions of CNN architecture to be tested are shown as binary combinations in Table I, where the pooling (2 × 2 subsampling) is indicated with ones, and zeros indicate that a ConvL is not followed by a pooling layer [9], [19], [20].

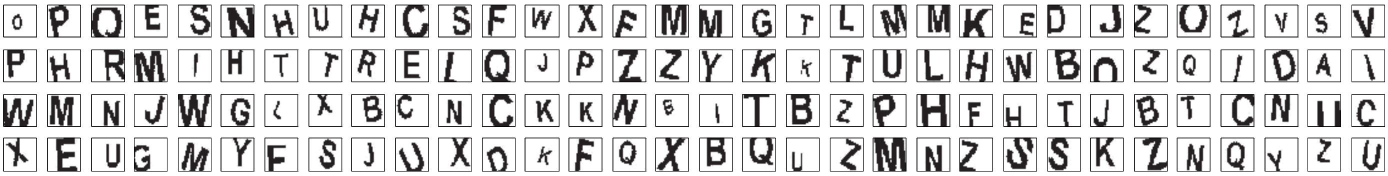

Fig. 3. A subset of the EEACL26 dataset consisting of 8-bit greyscale images created from originally monochrome 60×80 images [9]. Unlike CIFAR-10 or NORB, EEACL26 images are extremely simple; however, they fall into 26 classes.

TABLE I

VERSIONS OF CNN ARCHITECTURE TO BE TESTED ON THE DATASETS

#	CNN architecture (N_{ConvL})	Size of ConvLs' filters (in order of ConvLs numbering from the CNN input)	Image size (dimension)	Datasets
1	11 (2)	11, 10	32	CIFAR-10, NORB, EEACL26
2		17, 15	48	
3		21, 21	64	
4		33, 31	96	
5		37, 35	108	NORB
6	111 (3)	5, 5, 4	32	CIFAR-10, NORB, EEACL26
7		9, 7, 6	48	
8		9, 9, 9	64	
9		15, 14, 13	96	
10		17, 15, 15	108	NORB
11	1100 (4)	5, 5, 3, 3	32	CIFAR-10, NORB, EEACL26
12		7, 6, 5, 4	48	
13	1110 (4)	9, 7, 4, 4	64	
14		9, 9, 7, 6	96	
15	1111 (4)	9, 9, 8, 6	108	NORB
16	11010 (5)	5, 3, 2, 2, 2	32	CIFAR-10, NORB, EEACL26
17		5, 3, 3, 3, 3	48	
18	11101 (5)	5, 3, 3, 3, 3	64	
19	11111 (5)	5, 5, 4, 4, 2	96	
20		5, 5, 5, 3, 3	108	NORB
21	110010 (6)	3, 2, 2, 2, 2, 2	32	CIFAR-10, NORB, EEACL26
22	111000 (6)	5, 3, 3, 2, 2, 2	48	
23	111010 (6)	5, 5, 2, 2, 2, 2	64	
24	111101 (6)	5, 5, 4, 2, 2, 2	96	
25	111110 (6)	5, 5, 3, 2, 2, 2	108	NORB
26	1100100 (7)	3, 2, 2, 2, 2, 2, 1	32	CIFAR-10, NORB, EEACL26
27	1101000 (7)	5, 3, 2, 2, 2, 2, 2	48	
28	1110010 (7)	5, 5, 4, 2, 2, 2, 1	64	
29	1111010 (7)	5, 5, 4, 2, 2, 2, 1	96	
30		7, 6, 4, 3, 2, 2, 1	108	NORB
31	11000010 (8)	3, 2, 2, 2, 2, 2, 2, 1	32	CIFAR-10, NORB, EEACL26
32	11000100 (8)	5, 3, 2, 2, 2, 2, 2, 2	48	
33	11010010 (8)	5, 5, 3, 2, 2, 2, 2, 1	64	
34	11101000 (8)	5, 5, 2, 2, 2, 2, 2, 2	96	
35	11110010 (8)	5, 5, 3, 2, 2, 2, 2, 1	108	NORB
36	110000010 (9)	5, 3, 2, 2, 2, 2, 2, 2, 2	48	CIFAR-10, NORB, EEACL26
37	110100000 (9)	5, 3, 2, 2, 2, 2, 2, 2, 2	64	
38	110101000 (9)	5, 3, 2, 2, 2, 2, 2, 2, 2	96	
39	111010000 (9)	5, 3, 2, 2, 2, 2, 2, 2, 2	108	NORB
40	1010100000 (10)	3, 3, 2, 2, 2, 2, 2, 2, 2, 2	64	CIFAR-10, NORB, EEACL26
41	1101010000 (10)	5, 3, 2, 2, 2, 2, 2, 2, 2, 1	96	
42	1110100000 (10)	5, 3, 2, 2, 2, 2, 2, 2, 2, 1	108	NORB
43	11010001000 (11)	5, 3, 2, 2, 2, 2, 2, 2, 2, 2, 1	96	CIFAR-10, NORB, EEACL26
44	11100010000 (11)	5, 3, 2, 2, 2, 2, 2, 2, 2, 2, 1	108	NORB
45	110010010000 (12)	5, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1	108	

The listed architectures are close to being quasi-optimal for the corresponding N_{ConvL}. For accelerating the training processes, a single ReLU before the last ConvL is inserted, without DropOut layers [21], [22]. Although it would impair generalisation, our task is to obtain consistent statistics on performance. The performance consistency implies a good enough differentiation of error rate over various versions of CNN architecture (see Table I), which must help in finding the most appropriate integer(s) N_{ConvL}.

VI. Extraction of Integers N_{ConvL} Corresponding to the Best Performance

It takes a few epochs to obtain a sufficiently discriminated performance. Let $v_p^{\langle \mathrm{IRP} \rangle}(W, u)$ be the error rate for the IRP with image size $W \times W$ for the u-th CNN architecture version (the first column in Table 1) after the p-th epoch. Then the performance is normalised to either [9]

$$\tilde{v}^{\langle \mathrm{IRP} \rangle}(W, u) = \frac{\sum\limits_{p=1}^{8} v_p^{\langle \mathrm{IRP} \rangle}(W, u)}{\max\limits_{q \in Q_{\mathrm{IRP}}(W)} \sum\limits_{p=1}^{8} v_p^{\langle \mathrm{IRP} \rangle}(W, q)} \quad \text{by} \quad u \in Q_{\mathrm{IRP}}(W) \qquad (1)$$

or

$$\tilde{v}_8^{\langle \mathrm{IRP} \rangle}(W, u) = \frac{v_8^{\langle \mathrm{IRP} \rangle}(W, u)}{\max\limits_{q \in Q_{\mathrm{IRP}}(W)} v_8^{\langle \mathrm{IRP} \rangle}(W, q)} \quad \text{by} \quad u \in Q_{\mathrm{IRP}}(W) \qquad (2)$$

for comparing among IRPs, where $Q_{\mathrm{IRP}}(W)$ is the set of the versions for the given IRP and the given image size. For instance,

$$Q_{\text{CIFAR-10}}(32) = \{1, 6, 11, 16, 21, 26, 31\}$$

and

$$Q_{\text{CIFAR-10}}(96) = \{4, 9, 14, 19, 24, 29, 34, 38, 41, 43\}$$

are the sets for researching the minimum and maximum size of CIFAR-10 images. The sets are the same for EEACL26. The NORB dataset is researched in a wider range, starting with

$$Q_{\mathrm{NORB}}(32) = \{1, 6, 11, 16, 21, 26, 31\}$$

to

$$Q_{\mathrm{NORB}}(108) = \{5, 10, 15, 20, 25, 30, 35, 39, 42, 44, 45\}.$$

Figures 4–6 show the normalised error rates (1) polylined for fulfilling trend comparisons along the u axis, where $u \in \hat{Q}_{\text{CIFAR-10}}(W) \subset Q_{\text{CIFAR-10}}(W)$, $u \in \hat{Q}_{\mathrm{NORB}}(W) \subset Q_{\mathrm{NORB}}(W)$, $u \in \hat{Q}_{\mathrm{EEACL26}}(W) \subset Q_{\mathrm{EEACL26}}(W)$. The final-epoch normalised error rates (2) are polylined in Figures 7–9 by the same axes. A similarity between a dataset's polylines holds. However, the polylines of final-epoch-performance (2) look more scattered.

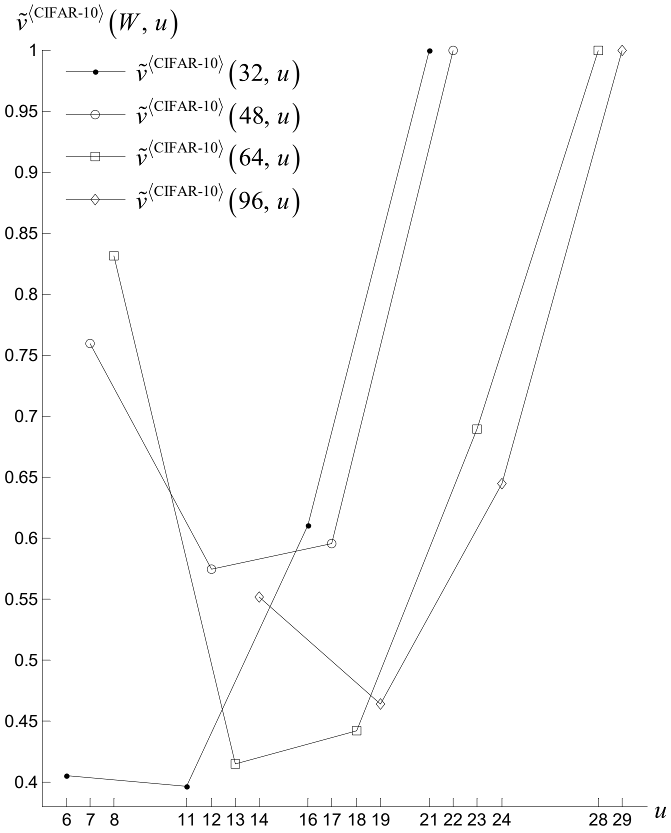

Fig. 4. The normalised error rates (1) for CIFAR-10. The best performance is observed at four ConvLs, except for the largest image size, for which the best performance corresponds to five ConvLs.

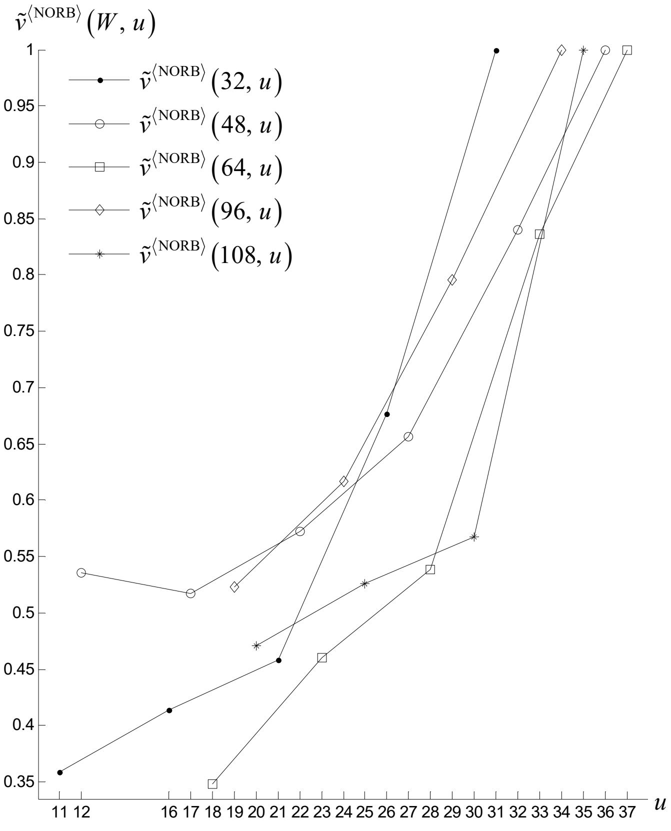

Fig. 5. The normalised error rates (1) for NORB. The best performance is observed at five ConvLs, except for the smallest image size, where the best performance is provided by four ConvLs.

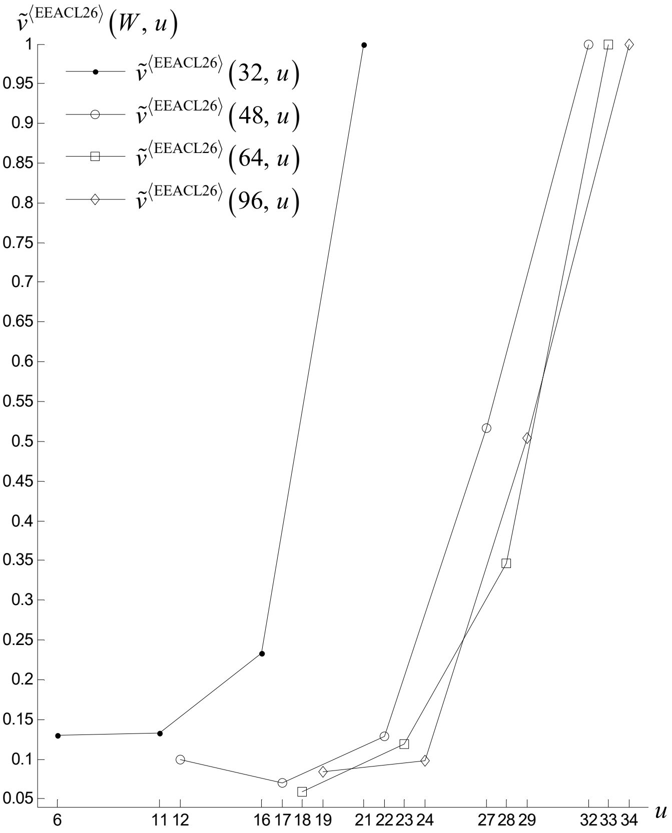

Fig. 6. The normalised error rates (1) for EEACL26. The best performance is observed at five ConvLs, except for the smallest images, where the best performance is provided by three or four ConvLs.

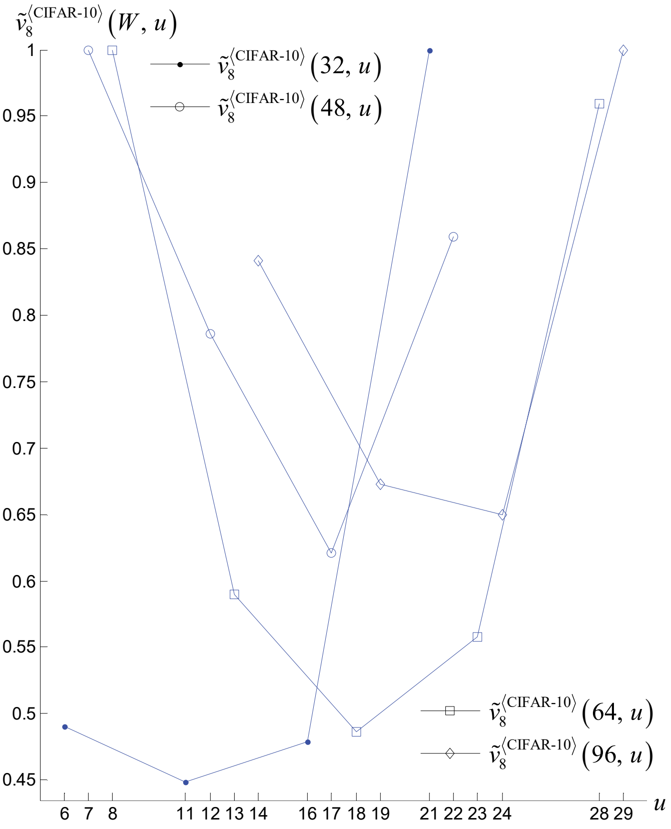

Fig. 7. The final-epoch normalised error rates (2) for CIFAR-10. The best N_{ConvL} for $W = 32$ is 4, the best N_{ConvL} for $W = 96$ is 6, $N_{\mathrm{ConvL}} = 5$ fits for the rest of the cases.

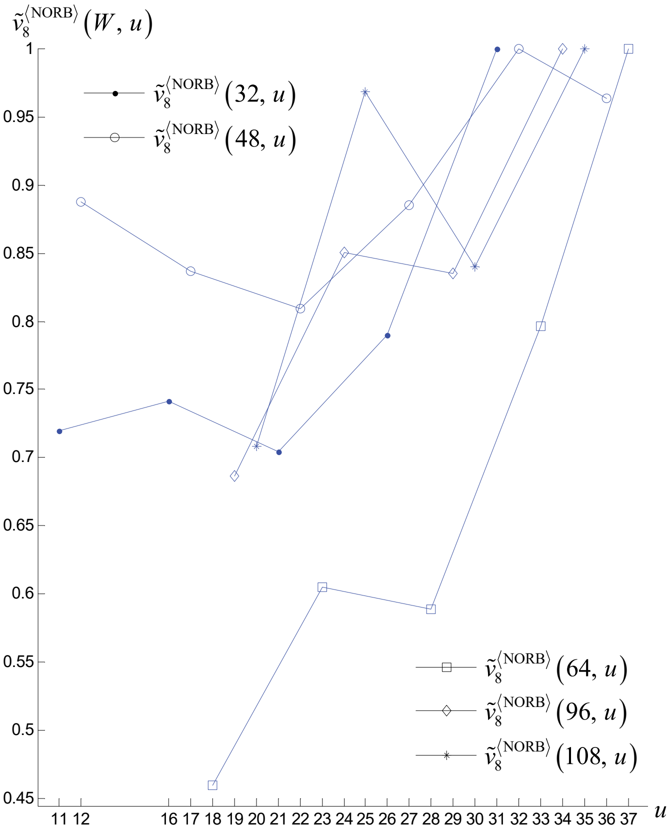

Fig. 8. The final-epoch normalised error rates (2) for NORB. Unexpectedly, $N_{\mathrm{ConvL}} = 5$ fits for $W \in \{64, 96, 108\}$ whereas the smaller images “prefer” $N_{\mathrm{ConvL}} = 5$.

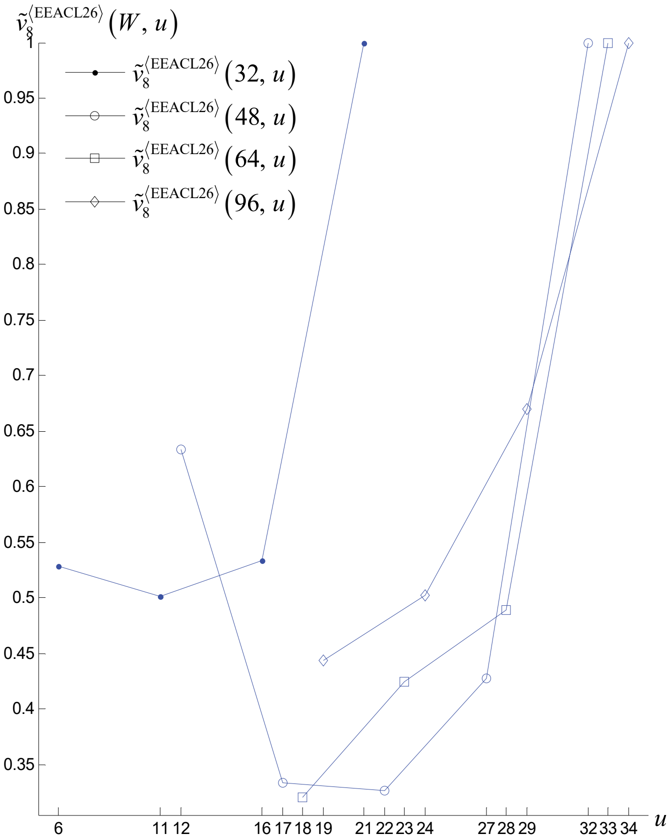

Fig. 9. The final-epoch normalised error rates (2) for EEACL26. For $W = 48$, two minima exist, so the appropriateness of ConvLs is similar to that in Fig. 6.

An apparent tendency that can be seen in Fig. 4–9 lies in the risk of CNN training failure when we increase the number of ConvLs. Too primitive architectures (consisting of only two ConvLs) do not work either. However, making a distinct conclusion on these polylines is hardly possible. So, further averaging is needed. This will not concern the size $W = 108$. As sets $\hat{Q}_{\text{CIFAR-10}}(W)$, $\hat{Q}_{\text{NORB}}(W)$, $\hat{Q}_{\text{EEACL26}}(W)$ are pairwise different (but, perhaps fortunately, not disjointed), the average performance of the three IRPs is to be viewed in the form (Fig. 10)

$$\tilde{v}(W, u) = \frac{\tilde{v}^{\langle\text{CIFAR-10}\rangle}(W, u) + \tilde{v}^{\langle\text{NORB}\rangle}(W, u) + \tilde{v}^{\langle\text{EEACL26}\rangle}(W, u)}{3} \quad (3)$$

and

$$\tilde{v}_8(W, u) = \frac{\tilde{v}_8^{\langle\text{CIFAR-10}\rangle}(W, u) + \tilde{v}_8^{\langle\text{NORB}\rangle}(W, u) + \tilde{v}_8^{\langle\text{EEACL26}\rangle}(W, u)}{3} \quad (4)$$

by

$$u \in \left\{\hat{Q}_{\text{CIFAR-10}}(W) \cap \hat{Q}_{\text{NORB}}(W) \cap \hat{Q}_{\text{EEACL26}}(W)\right\} \text{ for } W \in \{32, 48, 64, 96\}. \quad (5)$$

For the NORB dataset of the largest image size, formally,

$$\tilde{v}(108, u) = \tilde{v}^{\langle\text{NORB}\rangle}(108, u) \quad (6)$$

and

$$\tilde{v}_8(108, u) = \tilde{v}_8^{\langle\text{NORB}\rangle}(108, u) \quad (7)$$

by $u \in \hat{Q}_{\text{NORB}}(108)$. Data (6) and (7) being a segment "longer" than the rest, they are taken back from Figures 5 and 8, respectively.

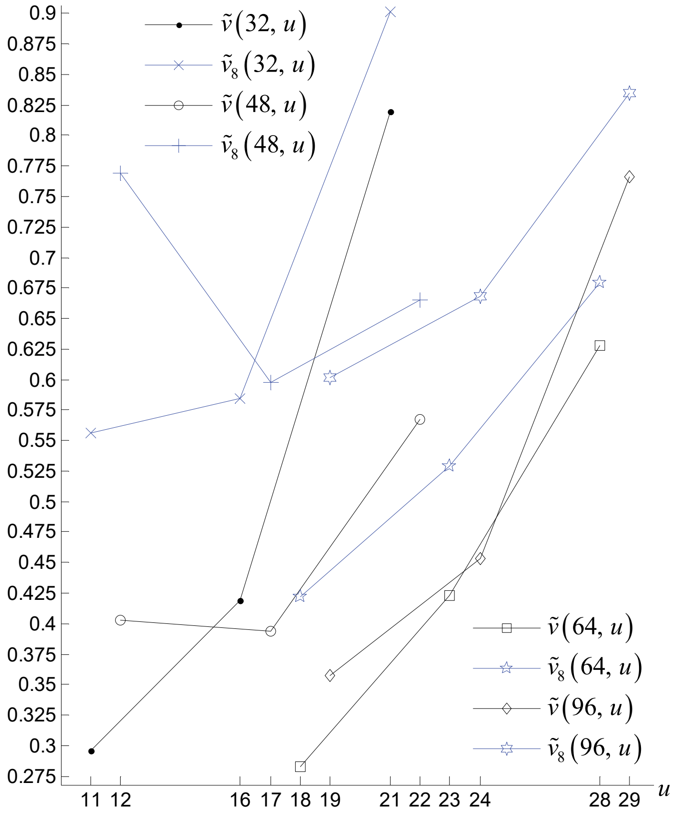

Fig. 10. The average performance of the three IRPs by (3) and (4), wherein only three common CNN architectures constitute an argument axis for each of the eight polylines. In the vertical direction, there are not more than two points above the same CNN architecture version. Except for the image size of 48, and 108 (only by final-epoch performance), all of these polylines (there are two-segmented lines, except for (6) consisting of three segments in Fig. 5) increase.

Although Figure 10 only deals with the dimensionality of an IRP, it gives us a straight conclusion on that IRPs of a higher dimensionality require more ConvLs. Nevertheless, the appropriate number of ConvLs for such IRPs is not much greater than that for lower dimensionalities: with the image size increased three times (from 32 up to 96), the appropriate N_{ConvL} does not change more than from 4 to 6 (if all the polylines are considered). Moreover, considering only the eight polylines in Figure 10, the appropriate N_{ConvL} is just 5 for any image size, except for 32×32 images, where the appropriate N_{ConvL} is 4 (see e.g. [9]).

VII. The Rule for an Appropriate N_{ConvL}

Apparently, as the image size increases, we may need more ConvLs. Then, however, the appropriate N_{ConvL} should always be slightly increased to prevent the risk of CNN training failure. Setting seven ConvLs for the benchmarked datasets has adverse consequences.

How does the number of image categories/classes influence the appropriateness of N_{ConvL} ? Table II, which contains integers N_{ConvL} that correspond to the error rate minima (in Figures 4–10) helps us see this. As can be easily seen, the dependence of the appropriate integer N_{ConvL} on the number of classes is hardly perceptible. It rather depends on the complexity of the IRP. And the number of classes is one of the components of the complexity of IRPs.

TABLE II

The Appropriate Number of ConvLs That Corresponds to the Error Rate Minima in Figures 4–10

	Datasets with the increasing numbers of classes					
	NORB		CIFAR-10		EEACL26	
W	Error rate (1)	Error rate (2)	Error rate (1)	Error rate (2)	Error rate (1)	Error rate (2)
32	4	6	4	4	4	4
48	5	6	4	5	5	6
64	5	5	4	5	5	5
96	5	5	5	6	5	5
108	5	5				

Hence, the rule for appropriate N_{ConvL} in CNNs is to try fewer ConvLs (an initial number) and then increase the number of ConvLs until the CNN performance starts deteriorating. For small images (like those in CIFAR-10), that initial number is 4. For much complex IRPs (in particular, ones with a few tens of image categories and more), it is recommended to initially set $N_{\text{ConvL}} = 5$. Definitely, the initial number of ConvLs for IRPs with a few thousand image categories is recommended to be set at 6, 7 or 8. Starting with $N_{\text{ConvL}} = 10$ is not recommended.

VIII. Conclusion

The attempt of finding an appropriate number of ConvLs in CNNs has been based on benchmarks of heterogeneous datasets. The heterogeneousness is principally needed for ensuring applicability to the appropriateness rule. Generally, the rule cannot give an exact number of ConvLs or even a few versions for this number outright. The rule is rather a short process of trying a few versions of N_{ConvL}, starting from $N_{\text{ConvL}} = 4$ for datasets whose image size is less than 100 and whose number of image categories is a few tens. In other cases, $N_{\text{ConvL}} \in \{5, 6, 7, 8\}$ at the beginning, where the greater N_{ConvL} corresponds to IRPs with a higher degree of complexity [23]. It seems that such fuzziness in the rule is not removable because of the required diversity and heterogeneousness of IRPs.

References

[1] H. H. Aghdam and E. J. Heravi, *Guide to Convolutional Neural Networks: A Practical Application to Traffic-Sign Detection and Classification*. Cham, Switzerland: Springer, 2017. https://doi.org/10.1007/978-3-319-57550-6

[2] A. Gibson and J. Patterson, *Deep Learning: A Practitioner's Approach*. O'Reilly Media, 2017.

[3] S. Srinivas, R. K. Sarvadevabhatla, K. R. Mopuri, N. Prabhu, S. S. S. Kruthiventi, and R. V. Babu, "Chapter 2 – An Introduction to Deep Convolutional Neural Nets for Computer Vision," in *Deep Learning for Medical Image Analysis*, S. K. Zhou, H. Greenspan, and D. Shen, Eds. Academic Press, 2017, pp. 25–52. https://doi.org/10.1016/b978-0-12-810408-8.00003-1

[4] V. Andrearczyk and P. F. Whelan, "Using Filter Banks in Convolutional Neural Networks for Texture Classification," *Pattern Recognition Letters*, vol. 84, pp. 63–69, Dec. 2016. https://doi.org/10.1016/j.patrec.2016.08.016

[5] Z. Liao and G. Carneiro, "A Deep Convolutional Neural Network Module that Promotes Competition of Multiple-Size Filters," *Pattern Recognition*, vol. 71, pp. 94–105, 2017. https://doi.org/10.1016/j.patcog.2017.05.024

[6] D. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, High Performance Convolutional Neural Networks for Image Classification," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, vol. 2, pp. 1237–1242, 2011.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification With Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, iss. 6, pp. 84–90, 2017. https://doi.org/10.1145/3065386

[8] J. Mutch and D. G. Lowe, "Object Class Recognition and Localization Using Sparse Features With Limited Receptive Fields," *International Journal of Computer Vision*, vol. 80, iss. 1, pp. 45–57, 2008. https://doi.org/10.1007/s11263-007-0118-0

[9] V. V. Romanuke, "Appropriate Number and Allocation of ReLUs in Convolutional Neural Networks," *Research Bulletin of the National Technical University of Ukraine "Kyiv Polytechnic Institute"*, no. 1, pp. 69–78, 2017. https://doi.org/10.20535/1810-0546.2017.1.88156

[10] P. Date, J. A. Hendler, and C. D. Carothers, "Design Index for Deep Neural Networks," *Procedia Computer Science*, vol. 88, pp. 131–138, 2016. https://doi.org/10.1016/j.procs.2016.07.416

[11] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *Computer Vision and Pattern Recognition*, 2015.

[12] V. V. Romanuke, "Boosting Ensembles of Heavy Two-Layer Perceptrons for Increasing Classification Accuracy in Recognizing Shifted-Turned-Scaled Flat Images With Binary Features," *Journal of Information and Organizational Sciences*, vol. 39, no. 1, pp. 75–84, 2015.

[13] V. V. Romanuke, "Two-Layer Perceptron for Classifying Flat Scaled-Turned-Shifted Objects by Additional Feature Distortions in Training," *Journal of Uncertain Systems*, vol. 9, no. 4, pp. 286–305, 2015.

[14] P. K. Rhee, E. Erdenee, S. D. Kyun, M. U. Ahmed, and S. Jin, "Active and Semi-Supervised Learning for Object Detection With Imperfect Data," *Cognitive Systems Research*, vol. 45, pp. 109–123, 2017. https://doi.org/10.1016/j.cogsys.2017.05.006

[15] P. Tang, H. Wang, and S. Kwong, "G-MS2F: GoogLeNet Based Multi-Stage Feature Fusion of Deep CNN for Scene Recognition," *Neurocomputing*, vol. 225, pp. 188–197, 2017. https://doi.org/10.1016/j.neucom.2016.11.023

[16] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper With Convolutions," *Computer Vision and Pattern Recognition*, 2014.

[17] V. V. Romanuke, "Classifying Scaled-Turned-Shifted Objects With Optimal Pixel-to-Scale-Turn-Shift Standard Deviations Ratio in Training 2-Layer Perceptron on Scaled-Turned-Shifted 4800-Featured Objects Under Normally Distributed Feature Distortion," *Electrical, Control and Communication Engineering*, vol. 13, iss. 1, pp. 45–54, 2017. https://doi.org/10.1515/ecce-2017-0007

[18] V. V. Romanuke, "Classification Error Percentage Decrement of Two-Layer Perceptron for Classifying Scaled Objects on the Pattern of Monochrome 60-by-80-Images of 26 Alphabet Letters by Training With Pixel-Distorted Scaled Images," *Scientific bulletin of Chernivtsi National University of Yuriy Fedkovych. Series: Computer systems and components*, vol. 4, iss. 3, pp. 53–64, 2013.

[19] M. Sun, Z. Song, X. Jiang, J. Pan, and Y. Pang, "Learning Pooling for Convolutional Neural Network," *Neurocomputing*, vol. 224, pp. 96–104, 2017. https://doi.org/10.1016/j.neucom.2016.10.049

[20] D. Scherer, A. Müller, and S. Behnke, "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition," in *International Conference on Artificial Neural Networks (ICANN 2010)*, pp. 92–101, 2010. https://doi.org/10.1007/978-3-642-15825-4_10

[21] S. Lai, L. Jin, and W. Yang, "Toward High-Performance Online HCCR: A CNN Approach With DropDistortion, Path Signature and Spatial Stochastic Max-Pooling," *Pattern Recognition Letters*, vol. 89, pp. 60–66, 2017. https://doi.org/10.1016/j.patrec.2017.02.011

[22] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks From Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[23] L. P. F. Garcia, A. C. P. L. F. de Carvalho, and A. C. Lorena, "Effect of Label Noise in the Complexity of Classification Problems," *Neurocomputing*, vol. 160, pp. 108–119, 2015. https://doi.org/10.1016/j.neucom.2014.10.085

Vadim V. Romanuke was born in 1979. The higher education was received in 2001. In 2006, he received the degree of Candidate of Technical Sciences in Mathematical Modelling and Computational Methods. His candidate dissertation suggested a way of increasing the interference noise immunity of data transferred over radio systems. Mr. Romanuke received his degree of Doctor of Technical Sciences in mathematical modelling and computational methods in 2014. His Doctor-of-Science dissertation solved the problem of increasing the efficiency of the identification of models for multistage technical control and run-in under multivariate uncertainties of their parameters and relationships. In 2016, he received the status of Full Professor.

Mr. Romanuke is a Professor at the Faculty of Navigation and Naval Weapons at the Polish Naval Academy. His research interests concern decision-making, game theory, statistical approximation, and control engineering based on statistical correspondence. Vadim Romanuke has good programming skills in MATLAB. For practical implementations, Mr. Romanuke uses Python. Also, he directs a branch of fitting statistical approximators at the Centre of Parallel Computations managed by Khmelnitskiy National University (Ukraine).

Address for correspondence: 69 Śmidowicza Street, Gdynia, Poland, 81–127.
E-mail: romanukevadimv@gmail.com
ORCID iD: https://orcid.org/0000-0003-3543-3087