DE GRUYTER OPEN
RIGA TECHNICAL UNIVERSITY
Applied Computer Systems
ISSN 2255-8691 (online)
ISSN 2255-8683 (print)
May 2018, vol. 23, no. 1, pp. 52–62
doi: 10.2478/acss-2018-0007
https://www.degruyter.com/view/j/acss

Smooth Non-increasing Square Spatial Extents of Filters in Convolutional Layers of CNNs for Image Classification Problems

Vadim V. Romanuke*
Polish Naval Academy, Gdynia, Poland

***Abstract* – The present paper considers an open problem of setting hyperparameters for convolutional neural networks aimed at image classification. Since selecting filter spatial extents for convolutional layers is a topical problem, it is approximately solved by accumulating statistics of the neural network performance. The network architecture is taken on the basis of the MNIST database experience. The eight-layered architecture having four convolutional layers is nearly best suitable for classifying small and medium size images. Image databases are formed of grayscale images whose size range is 28 × 28 to 64 × 64 by step 2. Except for the filter spatial extents, the rest of those eight layer hyperparameters are unalterable, and they are chosen scrupulously based on rules of thumb. A sequence of possible filter spatial extents is generated for each size. Then sets of four filter spatial extents producing the best performance are extracted. The rule of this extraction that allows selecting the best filter spatial extents is formalized with two conditions. Mainly, difference between maximal and minimal extents must be as minimal as possible. No unit filter spatial extent is recommended. The secondary condition is that the filter spatial extents should constitute a non-increasing set. Validation on MNIST and CIFAR-10 databases justifies such a solution, which can be extended for building convolutional neural network classifiers of colour and larger images.**

***Keywords* – Convolutional layer, convolutional neural networks, filters, hyperparameters, network architecture, square spatial extents of filters.**

I. An Open Problem of Setting Hyperparameters for Convolutional Neural Networks

Convolutional neural networks (CNNs) are a kind of statistical approximators intended for high-resolution image recognition. Encompassing both image classification and object detection, image recognition is a pivot of machine learning exploration and applications. As of October 2016, CNNs are the most popular deep learning architectures in computer vision [1]–[4].

A CNN architecture is formed by a stack of distinct layers that transform the input volume into the output volume. The transformation is executed through differentiable functions [5]–[7]. The output holds scores of the classes [8]–[10]. For creating CNN classifiers, a few distinct types of layers are commonly used [2], [3], [8], [9]. They are convolutional layers (ConvL), rectified linear unit layer (ReLU), average pooling layer (AvPL), max pooling layer (MaxPL), fully connected layer (FCL), softmax layer (SML) and dropout layer (DOL) [1]–[3], [8], [10], [11].

ConvL is a core building block of the CNN. A ConvL is a set of learnable filters (or kernels), which process portions of the input volume. These portions are called receptive fields [2], [9], [10]–[13]. A filter is commonly a $V \times H \times D$ matrix, where V is height of the filter (size by vertical axis), H is its width (size by horizontal axis), and D is the filter depth [2], [9], [11], [12]. The filter depth of the first ConvL is equal to the number of colour channels in the input image [11]. The filter depth of a successive ConvL is equal to the number of filters of the antecedent ConvL [8], [9], [11], [13]–[15]. If a total number of filters are K then a number of biases are K because each filter has a single bias [1], [2], [4], [8], [10], [11], [16]. Additionally, a stride and a pad are given along with a learning rate and weight decay [11], [17], [18].

The filter has a small receptive field, but extends through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the entries of the filter and the input. A separate 2-dimensional activation map of that filter is produced therein. Stacking the activation maps for all filters along the depth dimension forms the full output volume of the ConvL [2], [11], [16], [19], [20].

The size of the output volume of the ConvL is controlled by five hyperparameters [11], [12], [21]–[23]. They are V and H defining the size of the receptive field, depth of the output volume defined by K, the stride and zero-padding. For images having a lot of various oriented edges or blobs of colour, number K should be naively assigned greater [11], [24], [25]. Stride controls how depth columns around the spatial dimensions (width and height) are allocated. When the stride is 1, a new depth column of neurons is allocated to spatial positions only 1 spatial unit apart. This leads to ultimately overlapping receptive fields between the depth columns, producing ultimately large output volumes. The higher stride makes the receptive fields overlap less and the resulting output volume will have smaller dimensions [10], [11], [15], [17]. Zero-padding is used to exactly preserve the spatial size of output volumes. This helps easily aggregate the CNN architecture with integer V and H for all the filters in

* Corresponding author e-mail: romanukevadimv@gmail.com

ConvLs, and also with integer vertical and horizontal size of input volumes for AvPLs and MaxPLs. Besides, zero-padding preserving information about image borders may improve performance [10], [11], [16], [26], [27].

Those hyperparameters are assigned exploiting experience. However, the existing experience of CNNs is not enough for such a variety of image classification and object detection problems. Moreover, adjusting the ConvL hyperparameters is harder for recognising images of a bigger size.

II. Basic Research and Motivation

CNNs are very sensible to variation in their hyperparameters [8]–[11], [16], [21], [23]. Some heuristics allow setting them pretty effectively, but they concern narrow types of recognition problem [2], [3], [9], [28], [29]. There is no unified rule for setting CNN hyperparameters issuing from only image size and number of classes (categories).

Receptive fields are predominantly square, i.e., the spatial extent $F = V = H$ in all ConvLs [11], [16], [30]. Since feature map size decreases with CNN depth, ConvLs near the input tend to have fewer filters while ConvLs higher up have much more [11], [16], [31], i.e., the number of filters ordinarily increases for the next ConvLs. For preserving the information about the input image, the total number of activations (number of feature maps times number of pixel positions) is argued [1], [2], [8], [9], [11], [13]–[16], [18], [19], [25], [27], [28], [30], [32]–[34] to be non-decreasing from one layer to the next (or, severer, to be roughly constant across layers). Capacity of CNN is deeply connected with the number of feature maps. This number is also dependent on the number of available examples and the complexity of the problem.

As of November 2017, selection of the ConvL hyperparameters is mostly an art rather than science [2], [10], [11], [21], [24]. Particularly, the appropriate selection of filter size is a principal task for completing the CNN architecture. Configuration of ConvLs along with AvPLs, MaxPLs, ReLUs, FCLs, DOLs, SMLs in designing the CNN architecture is a much more difficult task. This task starts with the appropriately selected spatial extent of filters.

III. The Research Aims and Tasks to be Accomplished

Since selecting filter size for CNNs is an open topical problem, it should be approximately solved by some fixed CNN architecture. This architecture is expected to be roughly the best for a range of image classification problems. The numbers of filters will be constant. The other hyperparameters of CNN will be invariable as well. The criterion of selection is the CNN performance.

Tasks to be accomplished designate the research structure. They are the following:

1. To embody a range of the input image size in an image classification problem. The images should rather be as simple as possible for acquiring the research results faster.

2. To form an image database. The database must include three sets of images intended for training, validation and testing.

3. To substantiate a definite CNN architecture that is nearly best suitable for classifying images in the embodied range. Except for the filter spatial extents, the rest of CNN hyperparameters are unalterable, but they are chosen scrupulously based on rules of thumb.

4. To generate a sequence of possible filter spatial extents. Then, having evaluated the CNN performance, to find a subsequence that contains those filter spatial extents at which the performance is the best.

5. To formalize the rule of selecting the best filter spatial extents.

6. To validate the selected filter spatial extents on an appropriate benchmark. Validation sets will be done by changing the input image size within a range, including the range intended for the research before.

7. To discuss the research and validation results. A place of the suggested technique of adjusting the ConvL hyperparameters for the current CNNs is to be explicitly shown.

This list of the research tasks is a conceptual framework of solving an optimisation problem of CNNs. The solution is always near the optimality mode because the image range tasked to be used cannot encompass all the image forms and details. However, the framework is a CNN optimisation scheme, and it should be extended over other image classification problems requiring more complex CNN architectures.

IV. A Range of Size for Input Images to be Classified

Many explorations in CNNs use grayscale images [2], [8], [9], [16], [35], [36]. Even for RGB images in some datasets for classification, colour is revealed to be a low-level cue [37]. The image object identity in those datasets is invariant to changes in the intensity and colour of the illumination [11]. Therefore, images intended for the research will be grayscale.

Not considering the colour channels, input images are mostly resized to square form. Denote their size by $N \times N$. Based on the known datasets (databases) MNIST and CIFAR-10/100 [11], [16], [25], the range of the size is assigned 28×28 to 64×64. Let the step within the range be 2. Thus, the range of size for input grayscale images to be classified is

$$28 \times 28,\ 30 \times 30,\ 32 \times 32,\ ...,\ 62 \times 62,\ 64 \times 64. \qquad (1)$$

For acquiring the research results faster, let the image be the enlarged English alphabet capital letter (EEACL) [30], [38], [39]. Constituting 26 classes, these images are featured with horizontal and vertical lines, squares, circles, crossings, diagonals, curves, serpentine lines, etc. This is simultaneously simple and useful to accumulate CNN statistics in the fastest way [16], [38], [40], [41].

V. Database of Input Images for Training, Validation and Testing

An image database includes three sets of images intended for training, validation and testing. It is quite enough to have 2,000 images representing a class (Fig. 1). The training, validation and testing sets may be divided equally. In a general algorithmic way, a class pure representative is taken and distorted with scaling, rotation and shift [13], [38], [39].

After that, for a random variate Θ uniformly distributed on the unit half-interval with its values $\theta \in [0;1)$ and some points $\{\theta_{\text{train}}, \theta_{\text{val}}\}$ for which

$$0 < \theta_{\text{train}} < \theta_{\text{val}} < 1,$$

if

$$\theta < \theta_{\text{train}}, \tag{2}$$

then the current image is taken into the training set. Otherwise, if

$$\theta_{\text{train}} \leqslant \theta < \theta_{\text{val}}, \tag{3}$$

then the current image is taken into the validation set, and it is taken into the testing set by

$$\theta_{\text{val}} \leqslant \theta. \tag{4}$$

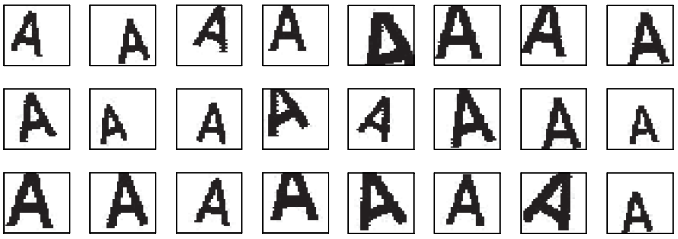

Fig. 1. The first class (category of the letter "A") is represented with 2,000 variously distorted 32×32 images [38]. The image background is pure white.

Normally, points

$$\{\theta_{\text{train}} = 1/3, \theta_{\text{val}} = 2/3\}$$

that makes the training, validation, and testing sets be approximately equal, or

$$\{\theta_{\text{train}} = 7/10, \theta_{\text{val}} = 17/20\} \tag{5}$$

or about that. Let us take (5) in our algorithm for making those sets, whereupon roughly a half of all images are included into the training set, and the rest of the database is roughly divided equally between validation and testing sets.

VI. CNN Architecture Based on the MNIST Database Classification Problem

CNNs have achieved an error rate of 0.23 % (as an ensemble of 35 networks) on the MNIST database [10], which as of February 2012 is the lowest one that has been achieved on the database having 10 classes (categories). The corresponding successful CNN architecture is

ConvL — MaxPL — ConvL — MaxPL — ConvL — ReLU — ConvL — SML

which is shortly written as

{ConvL — MaxPL}×2 — ConvL — ReLU — ConvL — SML. (6)

The MNIST database (Mixed National Institute of Standards and Technology database) is a large database of handwritten digits that is widely used for training and testing in the field of machine learning [8], [42], [43]. The MNIST database contains 60,000 training images and 10,000 testing images [42]. If necessary, it can be expanded into a set containing a few times more images using additional displacement (shift), scaling, rotation, etc. Usually, the MNIST image size is reshaped into 28×28 map. Hence, the architecture (6) is nearly best suitable for classifying images in the range (1).

The spatial extent F will be varied in each ConvL, and the rest of hyperparameters for (6) are chosen based on rules of thumb that allow achieving 0.23 % error rate on the MNIST database. In this way, the zero-padding is set to zero everywhere. The stride in all ConvLs is equal to 1. The stride in MaxPLs is 2, whose 2×2 filters perform, thus, a downsampling operation by 2 along the spatial dimensions, discarding 75 % of activations of the input volume.

Denote by K_l the number of filters in the l-th ConvL, $l = \overline{1, 4}$. Based on MNIST database classification problem, the numbers $\{K_l\}_{l=1}^{3}$ are set to the integers

$$\{20, 30, 100\}, \tag{7}$$

respectively. In the last (fourth) ConvL, $K_4 = 26$ accordingly to the number of classes of EEACLs in our database.

VII. Sequence of Possible Filter Spatial Extents

Having chosen those strides and zero-padding, the size of a 2-dimensional activation map at the output of the l-th ConvL is

$$W_l = N_l - F_l + 1 \quad \text{for} \quad l = \overline{1, 4} \tag{8}$$

by the size N_l of the input volume for the l-th ConvL and filter spatial extent F_l in this ConvL. The architecture (6) implies that the size of a 2-dimensional activation map of a filter in the first and second ConvLs is divisible by 2. Hence,

$$W_1 = N - F_1 + 1 \tag{9}$$

is the output size of the first ConvL and

$$N_2 = \frac{W_1}{2} \tag{10}$$

is the input size of the second ConvL. As number (10) must be integer, number (9) should be even. This is true for the range (1) along with an odd F_1. Thus, a minimal value of the filters' size of the first ConvL is

$$F_1 = 3 \tag{11}$$

and the maximal one is

$$F_1 = \frac{N}{2} + 1. \tag{12}$$

Obviously, the range of (11) to (12) is tried to be covered with the step equal 2.

The range of the filter size of the second ConvL is $F_2 = 2$ to $F_2 = N_2$ by (10). This range is tried to be covered with the step equal 1. As

$$N_3 = \frac{W_2}{2} \tag{13}$$

is the input size of the third ConvL by

$$W_2 = \frac{W_1}{2} - F_2 + 1, \tag{14}$$

then number (14) should be even. The range of the filter size of the third ConvL is $F_3 = 2$ to $F_3 = N_3$ by (13). Again, this range is tried to be covered with the step equal 1.

The input size of the fourth ConvL

$$\begin{aligned} N_4 = W_3 = N_3 - F_3 + 1 = \frac{W_2}{2} - F_3 + 1 = \\ = \frac{\frac{W_1}{2} - F_2 + 1}{2} - F_3 + 1 = \frac{W_1 - 2F_2}{4} - F_3 + \frac{3}{2} = \\ = \frac{N - F_1 + 1 - 2F_2}{4} - F_3 + \frac{3}{2} = \frac{N - F_1 - 2F_2}{4} - F_3 + \frac{7}{4} \end{aligned} \tag{15}$$

coincides with the filter spatial extent in this layer:

$$F_4 = \frac{N - F_1 - 2F_2}{4} - F_3 + \frac{7}{4}. \tag{16}$$

Thus, a set of filter spatial extents

$$\mathbf{S}_i(N) = \left(F_l^{\langle i \rangle}\right)_{1\times 4} \tag{17}$$

is made at the i-th iteration of the algorithm (8)–(16), where $F_l^{\langle i \rangle} = F_l$.

The set (sequence)

$$\boldsymbol{\mathcal{S}}(N) = \left\{\mathbf{S}_i(N) = \left(F_l^{\langle i \rangle}\right)_{1\times 4}\right\}_{i=1}^{B(N)} \tag{18}$$

of possible filter spatial extents calculated for $N \times N$ database is successively generated by the algorithm with formulae (8)–(16). The iteration starting points are (11), $F_2 = 2$, $F_3 = 2$. For instance,

$$\boldsymbol{\mathcal{S}}(28) = \{[3\ \ 2\ \ 2\ \ 5], [3\ \ 2\ \ 3\ \ 4], [3\ \ 2\ \ 4\ \ 3], \ldots, [15\ \ 4\ \ 2\ \ 1]\},$$

...,

$$\boldsymbol{\mathcal{S}}(48) = \{[3\ \ 2\ \ 2\ \ 10], \ldots, [25\ \ 7\ \ 2\ \ 2], [25\ \ 7\ \ 3\ \ 1], [25\ \ 9\ \ 2\ \ 1]\},$$

...,

$$\boldsymbol{\mathcal{S}}(64) = \{[3\ \ 2\ \ 2\ \ 14], [3\ \ 2\ \ 3\ \ 13], \ldots, [33\ \ 11\ \ 3\ \ 1], [33\ \ 13\ \ 2\ \ 1]\}.$$

The total number $B(N)$ of possible filter spatial extents increases as N increases. The increase resembles an exponential law (Fig. 2).

The CNN performance will be evaluated on each element of those 19 sequences. There are going 7,052 evaluations (CNNs) to be done altogether. For each sequence, a subsequence of filter spatial extents producing the best performance of CNN is expected to be. It would be ideal that those subsequences for all sizes in (1) have a common factor.

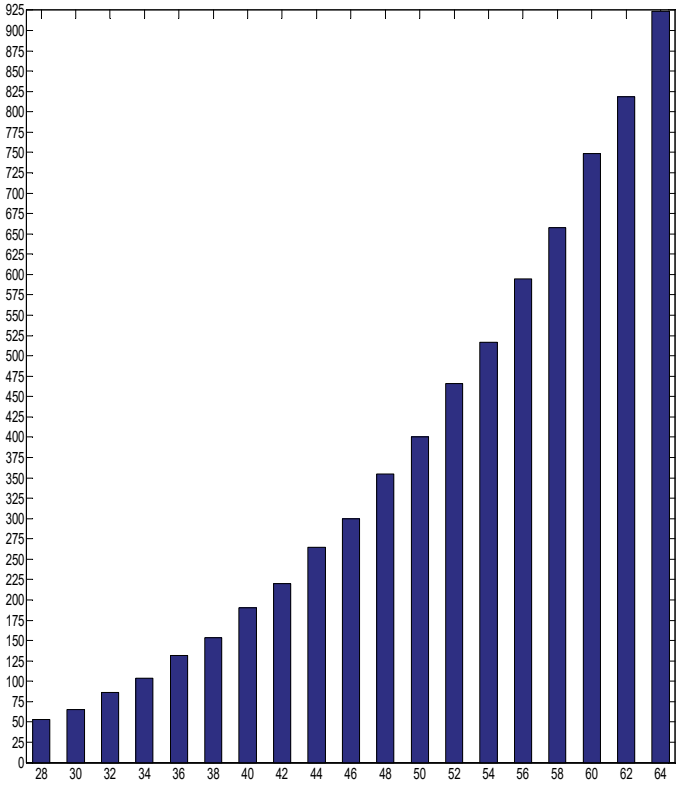

Fig. 2. Total number of possible filter spatial extents against the size of input images. The number increases having some semblance of an exponential law.

VIII. CNN Best Performance at a Subsequence of Filter Spatial Extents

Each of those 19 databases contains 52,000 $N \times N$ images of EEACLs. According to (5), nearly 70 % of images are used in training and 15 % of them are used in validating while training. The training consists of four epochs. The CNN is validated at each epoch. Thus, the CNN performance for the filter set (17) is represented with a four-element validation vector

$$\mathbf{V}_i(N) = \left(v_p^{\langle i \rangle}\right)_{1\times 4} \tag{19}$$

by an error rate percentage $v_p^{\langle i \rangle}$ after the p-th epoch. For point estimation, the CNN performance might be treated as

$$v_i(N) = \frac{1}{4}\sum_{p=1}^{4} v_p^{\langle i \rangle}. \tag{20}$$

However, the normed performance

$$\tilde{v}_i(N) = \frac{v_i(N)}{\max\left\{v_u(N)\right\}_{u=1}^{B(N)}} \tag{21}$$

is more convenient for comparison [44]–[46].

Having trained all 7,052 CNNs, there are 19 polylines obtained, whose vertices have abscissas labelled by 1 to $B(N)$ and ordinates $\{\tilde{v}_i(N)\}_{i=1}^{B(N)}$ for $N \times N$ image database (Fig. 3). When the image size is $N = 42$ and bigger, there are such combinations of filter spatial extents that the CNN training falls out of convergence. Number of such training-unfavourable combinations dramatically increases as N increases from 48 up to 64. They are mostly situated to the right of the abscissa axis, where integer F_1 increases. Even at this stage of analysis, this implies that selection of an exaggerated spatial extent of the first ConvL filter increases a likelihood of the total fail or hang. In some cases, derangement becomes not just after the first epoch, but after the second or later. These cases for $N \geqslant 48$ correspond to ordinates whose values are about between 0.3 and 0.9.

A common factor for those polylines in Fig. 3 is that some their parts are like a trough. It is seen that starting from the image size $N = 36$, these parts recur quasi-periodically.

Global troughs unifying local troughs are also seen. The "trough" subsequences are surely diverse, but those vertices whose ordinates are close to minimum are notable for their smoother distribution of filter spatial extents $\{F_l\}_{l=1}^{4}$. Specifically, no "trough" subsequence has the unit spatial extent.

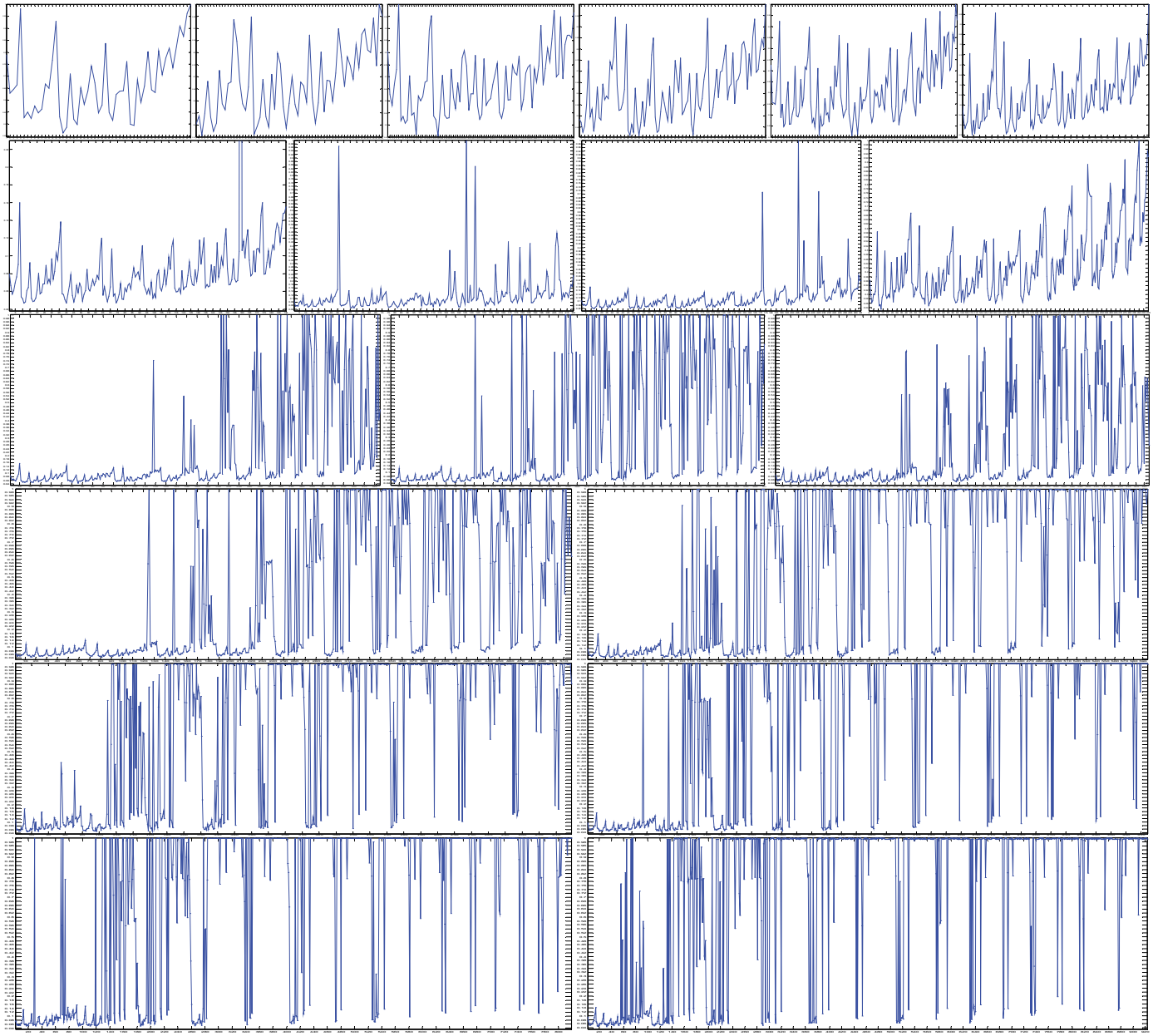

Fig. 3. The icon-like 19 polylines showing the normed averaged performance (21) against the ordered sequence of the filter spatial extents (to each image size).

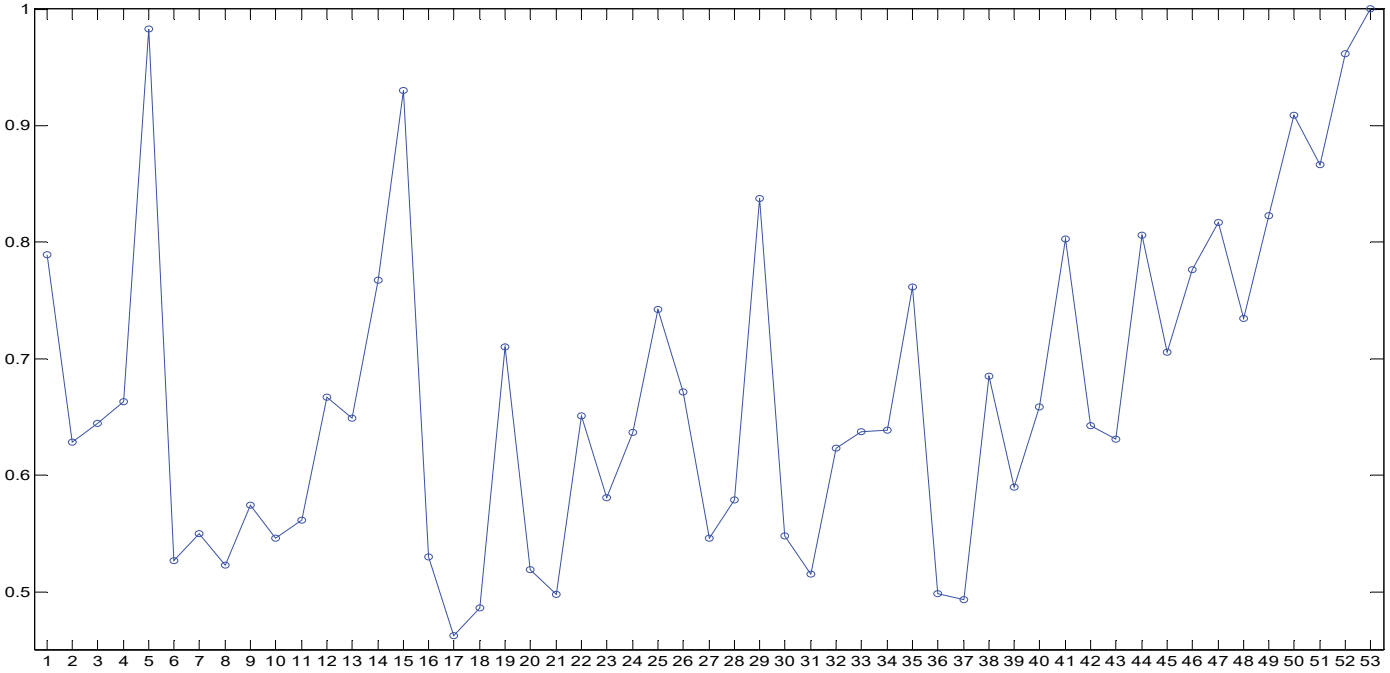

Fig. 4. The polyline showing the normed averaged performance (21) against the ordered sequence of the filter spatial extents for 28×28 images (from Fig. 3).

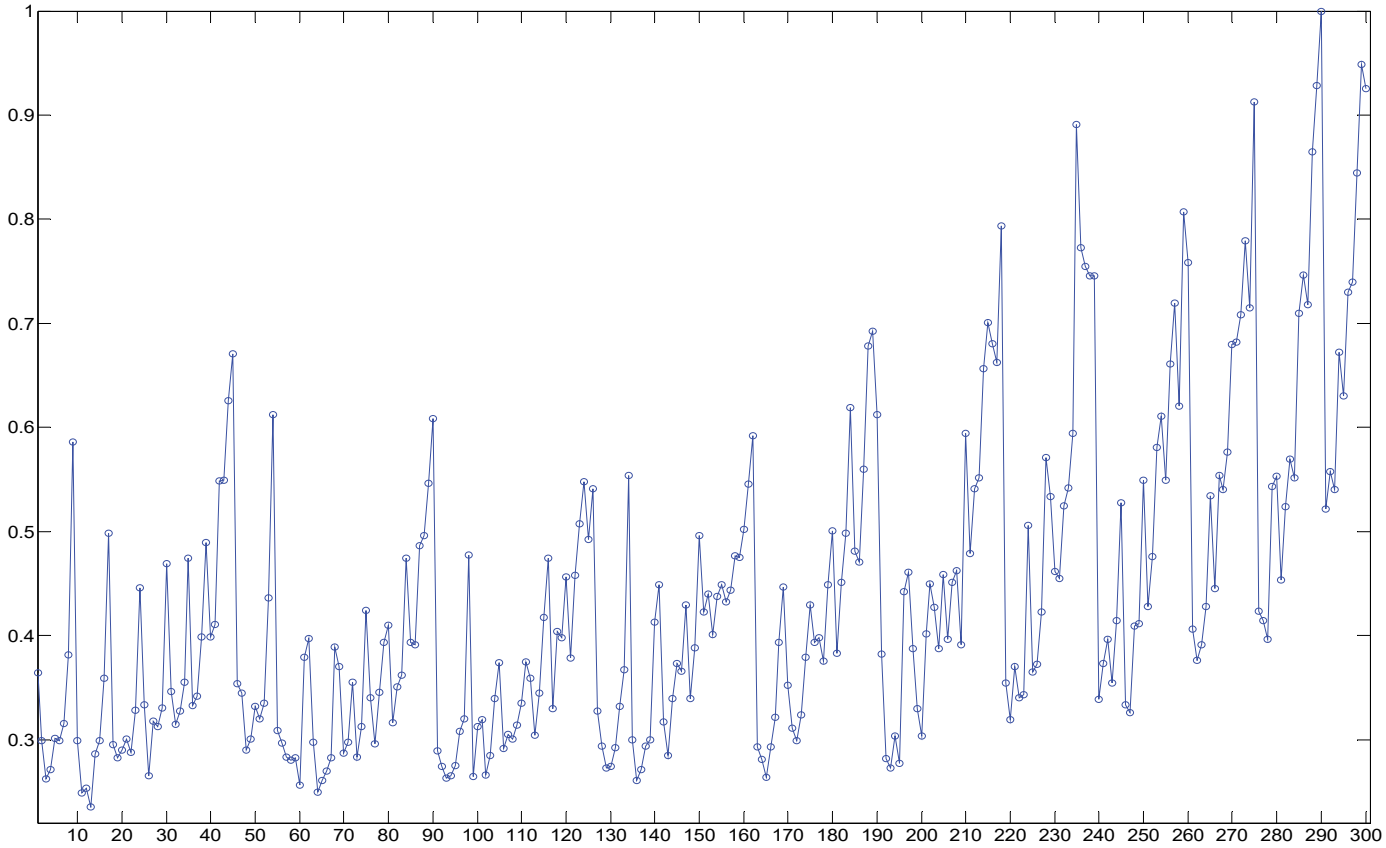

Fig. 5. The polyline showing the normed averaged performance (21) against the ordered sequence of the filter spatial extents for 46×46 images (from Fig. 3).

Another distinct feature of the "trough" subsequences is that distribution of filter spatial extents $\{F_l\}_{l=1}^4$ is roughly non-increasing. Besides, often sets $\{F_l\}_{l=1}^4$ of those subsequences contain equal integers starting from the second ConvL. This means the best performance of CNN is reached at filter spatial extents $\{F_l\}_{l=1}^4$ that are closer to satisfy the condition

$$F_1 \geqslant F_2 \geqslant F_3 \geqslant F_4 \quad \text{by} \quad \min\{F_l\}_{l=1}^4 > 1 \tag{22}$$

and the difference

$$\max\left\{F_l\right\}_{l=1}^{4}-\min\left\{F_l\right\}_{l=1}^{4} \tag{23}$$

taken as minimal as possible. But as the three inequalities (22) are hardly satisfied, the difference (23) is a host condition. Instead of (22), therefore, the number of cases of

$$F_k \geqslant F_l \quad \text{for} \quad k=\overline{1,3} \quad \text{and} \quad l=\overline{k+1,4} \quad \text{by} \quad \min\left\{F_l\right\}_{l=1}^{4}>1 \tag{24}$$

is counted up, and this number is to be rather great, up to 6.

The best filter spatial extents for 28×28 images is (see Fig. 4, which is a zoom of the first icon-like view from Fig. 3)

$$\mathbf{S}_{17}(28)=\begin{bmatrix}5 & 3 & 3 & 3\end{bmatrix} \tag{25}$$

producing the lowest error rate. The set (25) satisfies the conditions (22) and nearly minimizes (23). The best filter sizes for 46×46 images (Fig. 5)

$$\mathbf{S}_{13}(46)=\begin{bmatrix}3 & 5 & 5 & 5\end{bmatrix} \tag{26}$$

roughly satisfy them, making up 3 cases of (24). Among those sets of filter sizes, producing almost a minimal error rate, most sets nearly minimise (23) and maximise the number of cases of (24). This confirms that these conditions are crucial for setting filter spatial extents.

IX. Validation on MNIST and CIFAR-10 Databases

The set (25) is fit for training on the MNIST database, even without the training data expansion [2], [10], [11], [16], [43]. Only the filter numbers

$$\left\{K_1=20,\ K_2=50,\ K_3=500\right\}$$

are taken instead of the integers (7) that is expected to improve performance. Figure 6 shows comparison of the performance of CNN by the almost optimal set (25) and the "worst" set

$$\mathbf{S}_{53}(28)=\begin{bmatrix}15 & 4 & 2 & 1\end{bmatrix}$$

whose difference (23) is 14 and the last ConvL filters have unit spatial extent. The advantage of set (25) is clear making up about 2.85 times better performance before the derangement (after the 30-th epoch). The performance under set (25) continues its improving farther to the 128-th epoch.

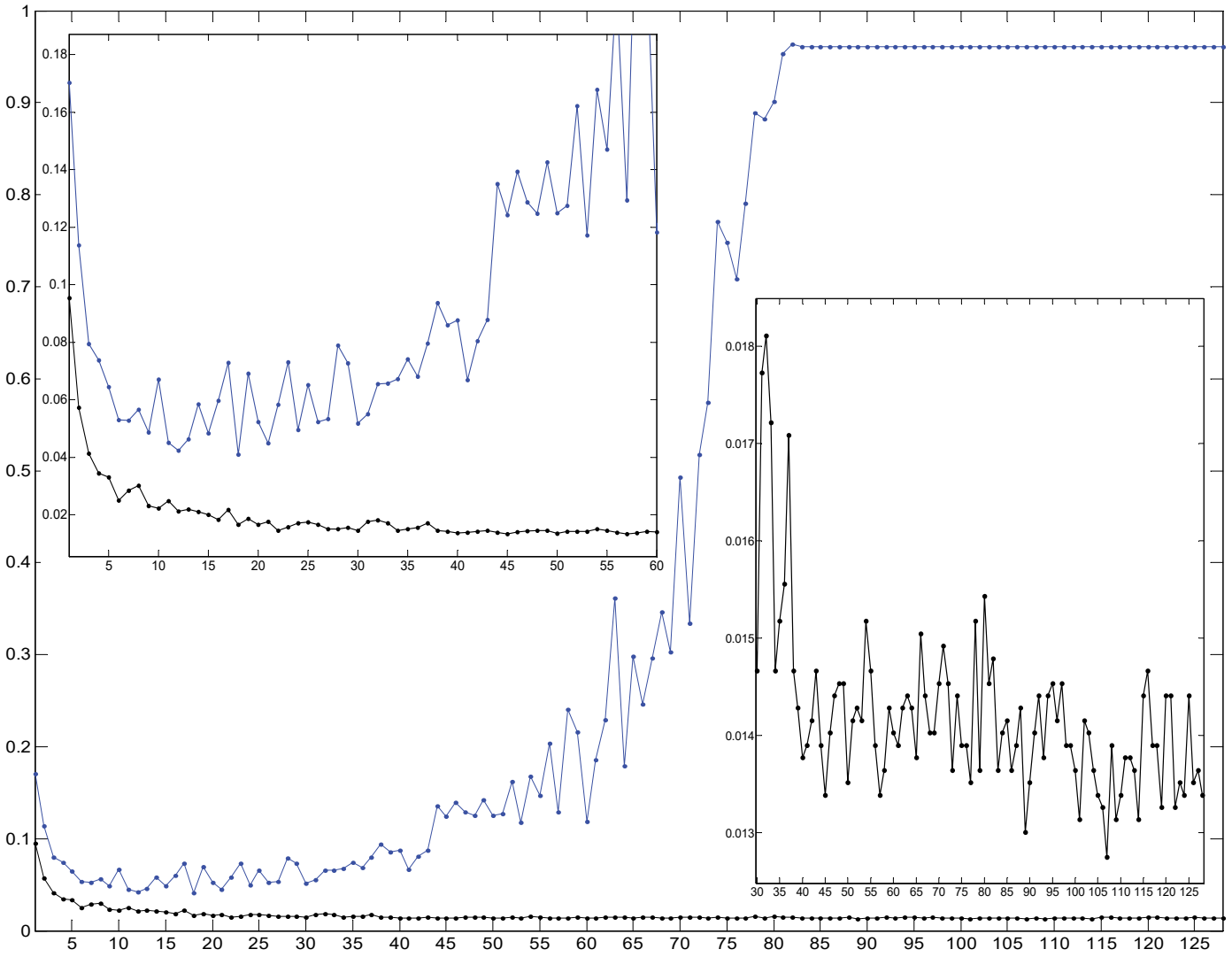

Fig. 6. The performance $\mathbf{V}_{17}(28)$ by (19) at $p=\overline{1,128}$ as the lower polyline against the upper polyline $\mathbf{V}_{53}(28)$ along 128 epochs (the "worst" effect is total).

Training on the CIFAR-10 database is much harder. It has also 10 classes, but its 32×32 images are more diverse. In fact, Figure 7 shows comparison of the CNN performance on the resized MNIST dataset by the almost optimal set

$$\mathbf{S}_{24}(32)=[5 \quad 3 \quad 4 \quad 3] \tag{27}$$

and the "worst" set

$$\mathbf{S}_{6}(32)=[3 \quad 2 \quad 7 \quad 1].$$

The advantage of set (27) is clear making up about 1.82 times better performance on average. In the same case, for CIFAR-10, along 16 epochs, the advantage of set (27) is 18 %.

Eventually, with a purpose to improve the latest error rate on the MNIST database, filter numbers

$$\{K_1=50,\ K_2=100,\ K_3=500,\ K_4=1000\}$$

are taken, implying an additional ConvL and a MaxPL by the additional four DOLs and four ReLUs. Filter spatial extents are set according to the stated conditions. Training for 74 epochs the seven corresponding CNNs (having 17 layers) on the expanded training data [27], [42], the result is 0.27 % error rate that outstrips the current result of the best single neural network (0.35 %). The best CNN obtained in such a way is used in boosting for reducing the MNIST error rate even more [43].

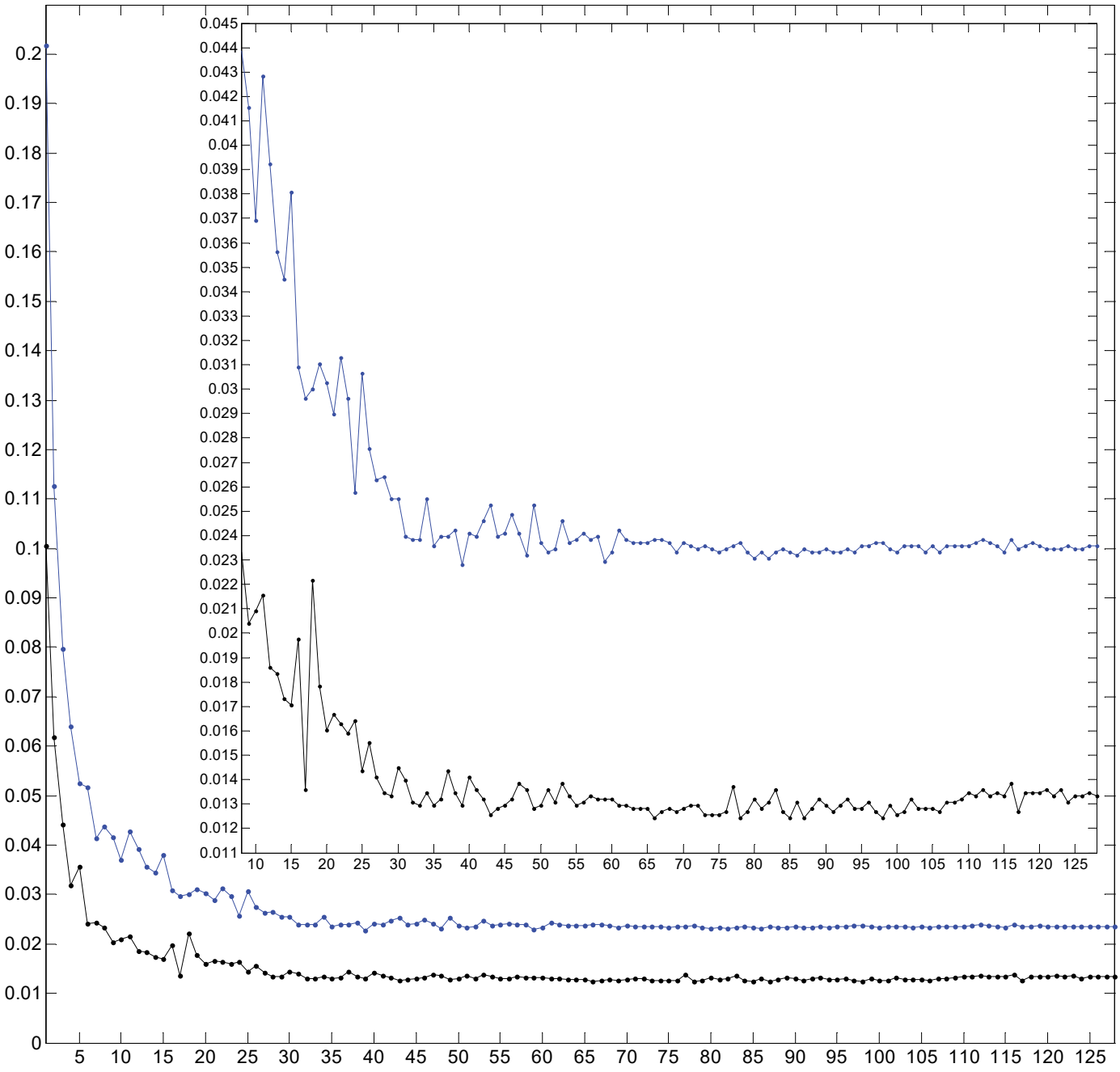

Fig. 7. The performance $\mathbf{V}_{24}(32)$ by (19) at $p=\overline{1,128}$ as the lower polyline against the upper polyline $\mathbf{V}_{6}(32)$ along 128 epochs. The "worst" effect is seen.

For further validation on the MNIST database, the input image size is changed from $N=28$ up to $N=128$. A distribution of filter spatial extents, which are nearly minimising (23) and maximising a number of cases of (24), is compared to a distribution where the filter size of the fourth ConvL is 1 (this one is the last in the ordered sequence). Every comparison result is similar to that one in Fig. 6 – the smoother "almost" non-increasing distribution outstrips the unsmoothed, although its sizes decrease (down to 1). Hypothetically, too sharp filter "focusing" is destructive.

X. Discussion

It is apparent that selection of a unit filter spatial extent is a poor practice (although it may happen for the ending ConvLs). That was just a reason for starting with (11), and not with $F_1=1$ that had been never seriously experienced before. For a CNN having L ConvLs, validation on MNIST and CIFAR-10 databases justifies the solution to the problem of selecting filter size by minimising difference

$$\max\left\{F_l\right\}_{l=1}^{L}-\min\left\{F_l\right\}_{l=1}^{L} \tag{28}$$

and maximising a number of cases of

$$F_k \geqslant F_l \quad \text{for} \quad k=\overline{1,\,L-1}$$
$$\text{and} \quad l=\overline{k+1,\,L} \quad \text{by} \quad \min\left\{F_l\right\}_{l=1}^{L}>1. \tag{29}$$

It appears perfect if this number is $\frac{L!}{2\cdot(L-2)!}$, which comes by

$$F_k \geqslant F_l \quad \text{at} \quad \min\left\{F_l\right\}_{l=1}^{L}>1. \tag{30}$$

Although architecture (6) is fixed, it seems that the same solution result holds for more complicated architectures having much more hyperparameters. It should also hold for colour and larger images.

The stated procedure is a quasi-optimisation framework based on statistical observations. It is self-evident that statistical approximators can be optimised by statistics (for an additional reference, see [39]). The optimum expressed here with the perfect distribution (30) and minimal difference (28) is approximate. One of its practical explanations is that non-increasing filter size fits best due to the fact that the size of activation map decreases with CNN depth. The decrement, if any, must not be abrupt.

The criterion of saving disk space and memory consumption was not considered. It would have made sense for CNN architectures intended to classify bigger images of a few hundred categories like the ImageNet database [11]. The quasi-optimisation directly on such a huge database is nonetheless unreasonable. The sense is just in the extension from a lighter and faster database classification problem, which actually is the range (1) for input grayscale images. Thus, the range (1) may be used for adjusting the ConvL hyperparameters for current CNNs in three steps. Firstly, the statistics is accumulated on a set of versions of the hyperparameter. Secondly, a subset of that set maximising the CNN performance (or, maybe, using other criterion) is extracted. After all, the subset selection is formalized similarly to that (28)–(30), allowing one to extend the quasi-optimum beyond.

XI. Conclusion

The present paper suggests a method of selecting optimally square receptive fields of ConvLs using statistical observations. The criterion is purely the CNN performance for as-smooth-as-possible non-increasing set of filter size. This is expressed with minimising difference (28), which is the host condition. Distribution (30) being the secondary condition is sometimes unattainable, so maximisation of cases of (29) is forced instead.

To apply this CNN optimisation scheme, a sequence of possible filter spatial extents is generated. From the very beginning, the sequence should not include unit spatial extents. Then a set of $\left\{F_l\right\}_{l=1}^{L}$ is selected for which difference (28) is minimal. It is likely that a few such sets can be selected, so a unique one is finally assigned for which the number of cases of (29) is maximal. For larger images, these rules may be violated. Thus, an appropriate set of filter spatial extents should be selected from the beginning of the sequence.

Further research must be tied with ascertaining how many filters at each ConvL are required for a definite classification problem. Moreover, strides and zero-padding are complementary open topical problems. Designing the CNN architecture, number and positions of ConvLs, ReLUs, MaxPLs (or AvPLs), DOLs, and FCLs are to be scientifically decided on.

References

[1] V. Chandrasekhar, J. Lin, O. Morère, H. Goh, and A. Veillard, "A practical guide to CNNs and Fisher Vectors for image instance retrieval," *Signal Processing*, vol. 128, 2016, pp. 426–439. https://doi.org/10.1016/j.sigpro.2016.05.021

[2] M. Elleuch, R. Maalej, and M. Kherallah, "A new design based-SVM of the CNN classifier architecture with dropout for offline Arabic handwritten recognition," *Procedia Computer Science*, vol. 80, 2016, pp. 1712–1723. https://doi.org/10.1016/j.procs.2016.05.512

[3] Q. Guo, F. Wang, J. Lei, D. Tu, and G. Li, "Convolutional feature learning and Hybrid CNN-HMM for scene number recognition," *Neurocomputing*, vol. 184, 2016, pp. 78–90. https://doi.org/10.1016/j.neucom.2015.07.135

[4] M. Joo Er, Y. Zhang, N. Wang, and M. Pratama, "Attention pooling-based convolutional neural network for sentence modelling," *Information Sciences*, vol. 373, 2016, pp. 388–403. https://doi.org/10.1016/j.ins.2016.08.084

[5] Z. Chen, F. Cao, and J. Hu, "Approximation by network operators with logistic activation functions," *Applied Mathematics and Computation*, vol. 256, 2015, pp. 565–571. https://doi.org/10.1016/j.amc.2015.01.049

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

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

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, iss. 11, 1998, pp. 2278–2324. https://doi.org/10.1109/5.726791

[9] P. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," *International Conference on Document Analysis and Recognition (ICDAR)*, vol. 3, 2003, pp. 958–962. https://doi.org/10.1109/ICDAR.2003.1227801

[10] D. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3642–3649. https://doi.org/10.1109/CVPR.2012.6248110

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, iss. 6, 2017, pp. 84–90. https://doi.org/10.1145/3065386

[12] 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, 2008, pp. 45–57. https://doi.org/10.1007/s11263-007-0118-0

[13] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, iss. 4, 1980, pp. 193–202. https://doi.org/10.1007/BF00344251

[14] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural Networks*, vol. 1, iss. 2, 1988, pp. 119–130. https://doi.org/10.1016/0893-6080(88)90014-7

[15] K. Fukushima, "Artificial vision by multi-layered neural networks: Neocognitron and its advances," *Neural Networks*, vol. 37, 2013, pp. 103–119. https://doi.org/10.1016/j.neunet.2012.09.016

[16] D. Ciresan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, vol. 2, 2011, pp. 1237–1242.

[17] P. Connor, P. Hollensen, O. Krigolson, and T. Trappenberg, "A biological mechanism for Bayesian feature selection: Weight decay and raising the LASSO", *Neural Networks*, vol. 67, 2015, pp. 121–130. https://doi.org/10.1016/j.neunet.2015.03.005

[18] A. Mahendran and A. Vedaldi, "Visualizing deep convolutional neural networks using natural pre-images," *International Journal of Computer Vision*, vol. 120, iss. 3, 2016, pp. 233–255. https://doi.org/10.1007/s11263-016-0911-8

[19] L. Guo, S. Li, X. Niu, and Y. Dou, "A study on layer connection strategies in stacked convolutional deep belief networks," *Pattern Recognition, 6th Chinese Conference, CCPR 2014, Changsha, China, November 17–19, 2014 (Proceedings, Part I)*, 2014, pp. 81–90. https://doi.org/10.1007/978-3-662-45646-0_9

[20] Z. Wang, Z. Deng, and S. Wang, "Accelerating convolutional neural networks with dominant convolutional kernel and knowledge pre-regression," *Computer Vision–ECCV 2016, 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII)*, 2016, pp. 533–548. https://doi.org/10.1007/978-3-319-46484-8_32

[21] Z.-Z. Li, Z.-Y. Zhong, and L.-W. Jin, "Identifying best hyperparameters for deep architectures using random forests," *Learning and Intelligent Optimization, 9th International Conference, LION 9, Lille, France, January 12–15, 2015 (Revised Selected Papers)*, 2015, pp. 29–42. https://doi.org/10.1007/978-3-319-19084-6_4

[22] C. Ann Ronao and S.-B. Cho, "Deep convolutional neural networks for human activity recognition with smartphone sensors," *Neural Information Processing, 22nd International Conference, ICONIP 2015, November 9–12, 2015 (Proceedings, Part IV)*, 2015, pp. 46–53. https://doi.org/10.1007/978-3-319-26561-2_6

[23] A. Azadeh, M. Saberi, A. Kazem, V. Ebrahimipour, A. Nourmohammadzadeh, and Z. Saberi, "A flexible algorithm for fault diagnosis in a centrifugal pump with corrupted data and noise based on ANN and support vector machine with hyper-parameters optimization," *Applied Soft Computing*, vol. 13, iss. 3, 2013, pp. 1478–1485. https://doi.org/10.1016/j.asoc.2012.06.020

[24] Z. Bai, L. L. C. Kasun, and G.-B. Huang, "Generic object recognition with local receptive fields based extreme learning machine," *Procedia Computer Science*, vol. 53, 2015, pp. 391–399. https://doi.org/10.1016/j.procs.2015.07.316

[25] P. Date, J. A. Hendler, and C. D. Carothers, "Design index for deep neural networks," *Procedia Computer Science*, vol. 88, 2016, pp. 131–138. https://doi.org/10.1016/j.procs.2016.07.416

[26] N. van Noord and E. Postma, "Learning scale-variant and scale-invariant features for deep image classification," *Pattern Recognition*, vol. 61, 2017, pp. 583–592. https://doi.org/10.1016/j.patcog.2016.06.005

[27] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *Computer Vision and Pattern Recognition*, arXiv:1312.6034v2 [cs.CV], 2014.

[28] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu, "Traffic sign detection and recognition using fully convolutional network guided proposals," *Neurocomputing*, vol. 214, 2016, pp. 758–766. https://doi.org/10.1016/j.neucom.2016.07.009

[29] J. Ma, F. Wu, J. Zhu, D. Xu, and D. Kong, "A pre-trained convolutional neural network based method for thyroid nodule diagnosis," *Ultrasonics*, vol. 73, 2017, pp. 221–230. https://doi.org/10.1016/j.ultras.2016.09.011

[30] J.-L. Buessler, P. Smagghe, and J.-P. Urban, "Image receptive fields for artificial neural networks," *Neurocomputing*, vol. 144, 2014, pp. 258–270. https://doi.org/10.1016/j.neucom.2014.04.045

[31] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *Computer Vision and Pattern Recognition*, arXiv:1506.06579v1 [cs.CV], 2015.

[32] L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks," *Computer Vision and Pattern Recognition*, arXiv:1505.07376v1 [cs.CV], 2015.

[33] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," *2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 3304–3311.

[34] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5188–5196. https://doi.org/10.1109/CVPR.2015.7299155

[35] C. Schmid and R. Mohr, "Local grayvalue invariants for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, iss. 5, 1997, pp. 530–535. https://doi.org/10.1109/34.589215

[36] V. Mayya, R. M. Pai, and M. M. M. Pai, "Automatic facial expression recognition using DCNN," *Procedia Computer Science*, vol. 93, 2016, pp. 453–461. https://doi.org/10.1016/j.procs.2016.07.233

[37] Y. LeCun, F. J. Huang, and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," *International Conference on Computer Vision and Pattern Recognition*, vol. 2, 2004, pp. 97–104. https://doi.org/10.1109/CVPR.2004.1315150

[38] 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, 2015, pp. 75–84.

[39] V. V. Romanuke, "Optimal training parameters and hidden layer neurons number of two-layer perceptron for generalized scaled objects classification problem," *Information Technology and Management Science*, vol. 18, 2015, pp. 42–48. https://doi.org/10.1515/itms-2015-0007

[40] 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, 2015, pp. 286–305.

[41] V. V. Romanuke, "An attempt for 2-layer perceptron high performance in classifying shifted monochrome 60-by-80-images via training with pixel-distorted shifted images on the pattern of 26 alphabet letters," *Radio Electronics, Computer Science, Control*, no. 2, 2013, pp. 112–118. https://doi.org/10.15588/1607-3274-2013-2-18

[42] E. Kussul and T. Baidyk, "Improved method of handwritten digit recognition tested on MNIST database," *Image and Vision Computing*, vol. 22, iss. 12, 2004, pp. 971–981. https://doi.org/10.1016/j.imavis.2004.03.008

[43] V. V. Romanuke, "Training data expansion and boosting of convolutional neural networks for reducing the MNIST dataset error rate," *Research Bulletin of the National Technical University of Ukraine "Kyiv Polytechnic Institute"*, no. 6, pp. 29–34, 2016. https://doi.org/10.20535/1810-0546.2016.6.84115

[44] V. V. Romanuke, "Uniform sampling of fundamental simplexes as sets of players' mixed strategies in the finite noncooperative game for finding equilibrium situations with possible concessions," *Journal of Automation and Information Sciences*, vol. 47, iss. 9, 2015, pp. 76–85. https://doi.org/10.1615/JAutomatInfScien.v47.i9.70

[45] V. V. Romanuke, "Sampling individually fundamental simplexes as sets of players' mixed strategies in finite noncooperative game for applicable approximate Nash equilibrium situations with possible concessions," *Journal of Information and Organizational Sciences*, vol. 40, no. 1, 2016, pp. 105–143.

[46] 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

Vadim V. Romanuke was born in 1979. He graduated from Technological University of Podillya in 2001. The higher education was received in 2001. In 2006, he received the Degree of Candidate of Technical Sciences in Mathematical Modelling and Computational Methods. The degree of Doctor of Technical Sciences in Mathematical Modelling and Computational Methods was received in 2014. In 2016, Vadim Romanuke received the academic status of Full Professor.

He is a Professor of the Faculty of Navigation and Naval Weapons at the Polish Naval Academy. His current research interests concern decision making, game theory, statistical approximation, and control engineering based on statistical correspondence. He is the author of 312 scientific articles, one monograph, one tutorial, three methodical guidelines in Functional Analysis, Mathematical and Computer Modelling, Conflict-Controlled Systems. At present, Vadim Romanuke is the scientific supervisor of a Ukrainian budget grant research concerning minimisation of water heat transfer and consumption. He also directs a branch of fitting statistical approximators at the Centre of Parallel Computations managed by Khmelnitskiy National University.

Address for correspondence: 69 Śmidowicza Street, Gdynia, Poland, 81-127.

E-mail: romanukevadimv@gmail.com