UDC 004.032.26:004.93

Romanuke V. V.

C. t. s., associate professor, Khmelnitskiy National University, Ukraine, E-mail: romanukevadimv@mail.ru

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

Object classification problem is considered, where neocognitron and multilayer perceptron may be applied. As neocognitron, solving almost any classification problem, performs too slowly and expensively, then for recognizing shifted monochrome images there is attempted the 2-layer perceptron, being fast only for pixel-distorted monochrome images, though. Having assumed the original images set of 26 monochrome 60-by-80-images of the English alphabet letters, there is formulated the task to clear out whether the 2-layer perceptron is capable to ensure high performance in classifying shifted monochrome images. Thus it is disclosed that the 2-layer perceptron performs as the good classifier of shifted monochrome images, when in training its input is fed with training samples from shifted images, being pixel-distorted. For this, however, it may need more passes of training samples through the 2-layer perceptron, but nevertheless the total traintime will be shorter than for training the 2-layer perceptron with only pixel-distortion-free shifted monochrome images.

Keywords: object classification, neocognitron, perceptron, pixel distortion, shift, monochrome images, shifted monochrome images.

PROBLEM OF SHIFT-TURN-SCALE PERCEPTRON RECOGNITION

Object recognition is an up-to-date technical problem, dealing with a lot of aspects in its formalization and solving it. The mathematical principle for recognizing objects lies in clustering and classifying them. Neural network, being the universal approximator, is the finest model of clusterization and classification [1, 2]. It needs neither architecture creation, nor training algorithm development. Only the appropriate architecture must be selected among the available ones [1, 3], and the corresponding training algorithm should be enabled [4, 5]. There nonetheless stands an important question of ensuring the high productivity up with low resources consumption and short response delay. The highest productivity is ensured by neocognitron, which is the smartest neural network, performing slowly, though [6, 7]. It also takes too much of memory and data space for clusterization and classification. The multilayer perceptron, consuming memory not so significantly, works much faster, but its productivity is ensured high only for objects or noised objects, which are not shifted, turned (skewed) or scaled against the training objects sample [8]. Certainly, this is unreal situation in world of real events and processes. And if the object, being under classification, is shifted, turned or scaled against its original in the training sample, then perceptron cannot recognize it and classifies this object erroneously. Therefore the problem of shift-turn-scale (STS) object recognition may be solved with making neocognitron perform easier and faster or with making perceptron just recognize STS-property objects better.

WAY OF INVESTIGATION

Clearly that both the said tasks of constructing fast neocognitron and training perceptron STS-classifier are tough for implementation [4, 6, 7]. Besides, neocognitron is very huge model to try optimizing it in speed and resources consumption. So, there is a tenable way of trying to prepare multilayer perceptrons for classifying objects just with one from the three STS-properties: shift, turn or scale. The shift-property is the easiest to program it, while turn or scale is tougher to be modeled [6]. The simplest objects are plane objects like monochrome images. However, even for shifted monochrome images (SMI) multilayer perceptrons perform poorly. Although, multilayer perceptrons perform well [4, 8] for pixel-noised monochrome images (PNMI). It outlines the way to investigate possibility of increasing multilayer perceptron performance in classifying SMI, whether they are PNMI or not.

TASK FORMULATION

Will be considering the original set of 26 monochrome 60-by-80-images which are 26 letters of the English alphabet. An alphabet letter, feeding the classifier input, may be shifted as it occurs usually while scanning and retrieving the text information. The classifier must recognize it at high performance. Cases with letters, being PNMI, are not excluded, but the main case is that input objects are SMI. The extent of shift is going to be featured with a shift constant. The task is to clear out whether the 2-layer perceptron (2LP) is capable to ensure high performance for SMI. For that the model of PNMI and shift-noised

monochrome images (SNMI) must be formalized, whereupon 2LP is trained to become the classifier.

MODEL OF PNMI

It is known that 2LP is trained with blocks of feature-vectorized objects, so q-th image as matrix $\mathbf{A}_q = \left(a_{uv}^{\langle q \rangle}\right)_{60\times 80}$ is reshaped into 4800-length-column before processing it, where $a_{uv}^{\langle q \rangle} \in \{0, 1\}$ by $q = \overline{1, 26}$. Denote $\mathbf{A} = \left(\overline{a}_{jq}\right)_{4800\times 26}$ as the matrix of all original 26 monochrome 60-by-80-images, reshaped into 26 columns. Then model of PNMI is just the matrix

$$\mathbf{A}_{\text{pixel}}^{\langle k \rangle} = \mathbf{A} + \sigma_{\text{pixel}}^{\langle k \rangle} \cdot \mathbf{\Xi} \tag{1}$$

by standard deviation

$$\sigma_{\text{pixel}}^{\langle k \rangle} = \frac{\sigma_{\text{pixel}}^{\langle \max \rangle}}{F} \cdot k \quad \forall\ k = \overline{1, F} \tag{2}$$

and its maximum $\sigma_{\text{pixel}}^{\langle \max \rangle} > 0$ at 4800×26-matrix $\mathbf{\Xi}$ of values of normal variate with zero expectation and unit variance, where number F indicates at smoothness in training the perceptron [9]. While being trained, the input of 2LP is fed with the set

$$\tilde{P}_{\text{train}} = \left\{\tilde{\mathbf{P}}_i\right\}_{i=1}^{C+F} = \left\{\left\{\mathbf{A}\right\}_{l=1}^{C}, \left\{\mathbf{A}_{\text{pixel}}^{\langle k \rangle}\right\}_{k=1}^{F}\right\} \tag{3}$$

of original images and pixel-distorted images by the set of identifiers (targets)

$$T = \left\{\mathbf{T}_i\right\}_{i=1}^{C+F} = \left\{\mathbf{I}\right\}_{i=1}^{C+F} \tag{4}$$

with identity 26×26-matrix $\mathbf{I}$, where number C indicates at how many replicas of undistorted images should be recognized in the training process. The set (3), being formed by (1) and (2), is passed through 2LP with identifiers (4) for Q_{pass} times.

One of the fastest program implementations of 2LP can be built within MATLAB with using the training function «traingda». For setting the size of hidden layer of 2LP to 250 neurons at $\sigma_{\text{pixel}}^{\langle \max \rangle} = 1$, $C = 2$, $F = 8$, $Q_{\text{pass}} = 10$, the results of classifying SMI, when 2LP is trained with PNMI, appear quite unacceptable (figure 1). These results are obtained in routine of the batch testing of 2LP. The results of the letter-by-letter testing of 2LP at some fixed standard deviation for (1) disclose the trend in distribution of recognition errors percentage over letters (figure 2). This trend is seen that there exist letters that are recognized better than others. For instance, letters «I» and «J» are more recognizable, where letter «I» is classified wrong only in every fourth case, roughly. But the averaged recognition errors percentage nonetheless remains inadmissibly high.

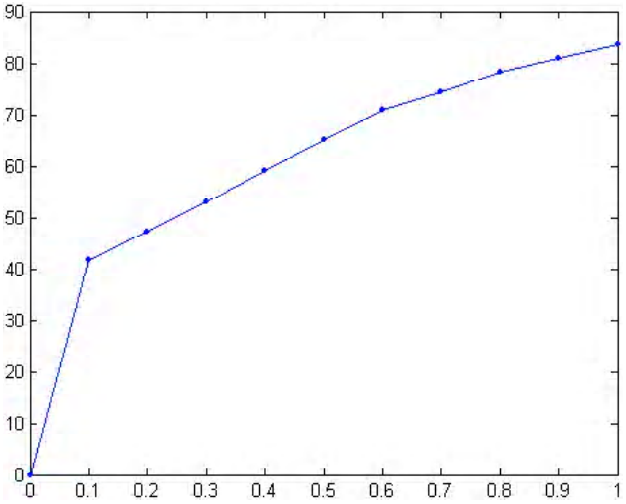

Fig. 1. Percentage of recognition errors p_{error} over standard deviation range $\left[0;\ \sigma_{\text{shift}}^{\langle \max \rangle}\right]$ of the shift intensity in SMI by 250 neurons in 2LP hidden layer and $\sigma_{\text{shift}}^{\langle \max \rangle} = 1$, $\sigma_{\text{pixel}}^{\langle \max \rangle} = 0$ (derived from 1000 batch testings of the trained 2LP with PNMI by $\sigma_{\text{pixel}}^{\langle \max \rangle} = 1$, $C = 2$, $F = 8$, $Q_{\text{pass}} = 10$)

MODEL OF SNMI

Just like in (1), model of PNMI consists in adding the normal noise to the matrix of all 26 images. For one from the three STS-properties, every image should be processed separately. In model of SNMI each image is shifted horizontally and vertically for some number of pixels. Thus the shift constant is from two components, horizontal and vertical, though there may be used the same standard deviation

$$\sigma_{\text{shift}}^{\langle k \rangle} = \frac{\sigma_{\text{shift}}^{\langle \max \rangle}}{F} \cdot k \quad \forall\ k = \overline{1, F} \tag{5}$$

and $\sigma_{\text{shift}}^{\langle \max \rangle} > 0$ at k-th part of forming the set that feeds the input of 2LP. As there are considered 60-by-80-images then horizontal pixel shift (HPS) is

$$s_{\text{hor}}\left(\sigma_{\text{shift}}^{\langle k \rangle}\right) = \varphi\left(8\sigma_{\text{shift}}^{\langle k \rangle} \cdot \xi_{\text{hor}}(k)\right) \times \frac{1 - \text{sign}\left(\left|\varphi\left(8\sigma_{\text{shift}}^{\langle k \rangle} \cdot \xi_{\text{hor}}(k)\right)\right| - 80\right)}{2} + 80 \cdot \frac{1 + \text{sign}\left(\left|\varphi\left(8\sigma_{\text{shift}}^{\langle k \rangle} \cdot \xi_{\text{hor}}(k)\right)\right| - 80\right)}{2}, \tag{6}$$

where $\xi_{\text{hor}}(k)$ is value of normal variate with zero expectation and unit variance, raffled at the k-th stage for

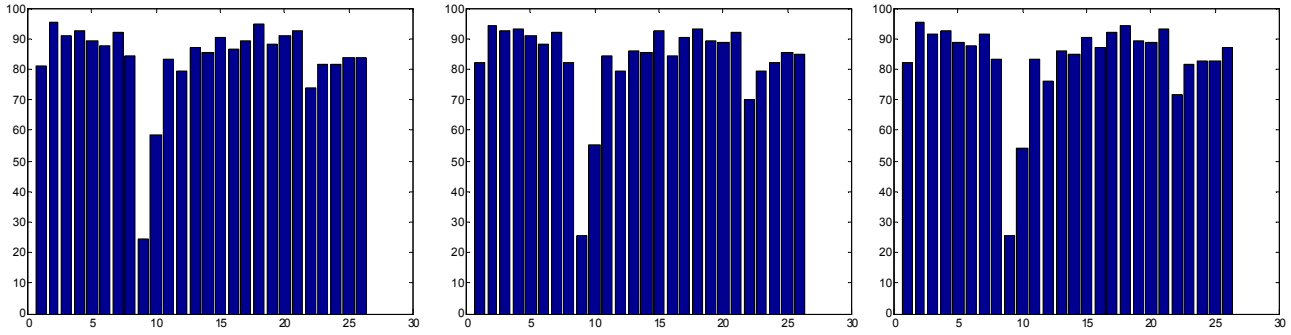

Fig. 2. Distributions of recognition errors percentage over letters at fixed standard deviation $\sigma_{\text{shift}}^{\langle\max\rangle}=1$ for $\sigma_{\text{pixel}}^{\langle\max\rangle}=0$ (derived from 1000 testings of the PNMI-trained-and-loaded 2LP at $\sigma_{\text{pixel}}^{\langle\max\rangle}=1$, $C=2$, $F=8$, $Q_{\text{pass}}=10$)

HPS, and function $\varphi(x)$ rounds x to the nearest integer less than or equal to x. Concurrently, vertical pixel shift (VPS) is

$$s_{\text{ver}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)=\varphi\left(6\sigma_{\text{shift}}^{\langle k\rangle}\cdot\xi_{\text{ver}}(k)\right)\times$$
$$\times\frac{1-\text{sign}\left(\left|\varphi\left(6\sigma_{\text{shift}}^{\langle k\rangle}\cdot\xi_{\text{ver}}(k)\right)\right|-60\right)}{2}+$$
$$+60\cdot\frac{1+\text{sign}\left(\left|\varphi\left(6\sigma_{\text{shift}}^{\langle k\rangle}\cdot\xi_{\text{ver}}(k)\right)\right|-60\right)}{2},\qquad(7)$$

where $\xi_{\text{ver}}(k)$ is value of normal variate with zero expectation and unit variance, raffled at the k-th stage for VPS.

It is necessarily to mind that the image background is white, whereas in MATLAB the white color is coded with ones. So, contour and filling of letters, being black, are coded with zeros. By the way, the filling is not continuous (figure 3), and the letter black cast is sprinkled with white specks. Hence, adding the horizontal shift noise to q-th image as matrix $\mathbf{A}_q=\left(a_{uv}^{\langle q\rangle}\right)_{60\times80}$ changes its elements into the following. For $s_{\text{hor}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)>0$ these new elements are

$$\tilde{a}_{uv}^{\langle q\rangle}(k)=1 \text{ for } u=\overline{1,\,60} \text{ and } v=\overline{1,\,s_{\text{hor}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)}\qquad(8)$$

by

$$\tilde{a}_{uv}^{\langle q\rangle}(k)=a_{ut}^{\langle q\rangle} \text{ at } t=v-s_{\text{hor}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)$$

$$\text{for } u=\overline{1,\,60} \text{ and } v=\overline{s_{\text{hor}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)+1,\,80}.\qquad(9)$$

For $s_{\text{hor}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)<0$ new elements are

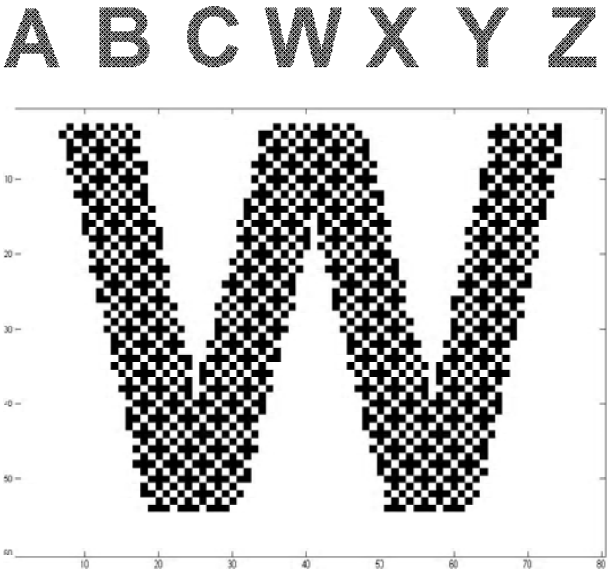

Fig. 3. Monochrome 60-by-80-images of letters, viewed as bitmap files, and the letter «W», viewed within MATLAB, where it can be seen that the letter black cast is sprinkled crosshatch-regularly with white specks

$$\tilde{a}_{uv}^{\langle q\rangle}(k)=a_{ut}^{\langle q\rangle} \text{ at } t=v-s_{\text{hor}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)$$

$$\text{for } u=\overline{1,\,60} \text{ and } v=\overline{1,\,80+s_{\text{hor}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)}\qquad(10)$$

by

$$\tilde{a}_{uv}^{\langle q\rangle}(k)=1 \text{ for}$$

$$u=\overline{1,\,60} \text{ and } v=\overline{80+s_{\text{hor}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)+1,\,80}.\qquad(11)$$

Clearly, for $s_{\text{hor}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)=0$ the q-th image is not shifted horizontally:

$$\tilde{a}_{uv}^{\langle q\rangle}(k)=a_{uv}^{\langle q\rangle} \text{ for } u=\overline{1,\,60} \text{ and } v=\overline{1,\,80}.\qquad(12)$$

After that horizontal shift is accomplished, adding the vertical shift noise to the horizontally shifted q-th image as matrix $\tilde{\mathbf{A}}_q(k)=\left[\tilde{a}_{uv}^{\langle q\rangle}(k)\right]_{60\times 80}$ changes its elements into the following. For $s_{\text{ver}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)>0$ these refreshed elements are

$$\tilde{\tilde{a}}_{uv}^{\langle q\rangle}(k)=\tilde{a}_{rv}^{\langle q\rangle}(k) \text{ at } r=u+s_{\text{ver}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)$$

$$\text{for} \quad u=\overline{1,\ 60-s_{\text{ver}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)} \text{ and } v=\overline{1,\ 80} \qquad (13)$$

by

$\tilde{\tilde{a}}_{uv}^{\langle q\rangle}(k)=1$ for

$$u=\overline{60-s_{\text{ver}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)+1,\ 60} \text{ and } v=\overline{1,\ 80}. \qquad (14)$$

For $s_{\text{ver}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)<0$ new elements are

$$\tilde{\tilde{a}}_{uv}^{\langle q\rangle}(k)=1 \text{ for } u=\overline{1,\ -s_{\text{ver}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)} \text{ and } v=\overline{1,\ 80} \qquad (15)$$

by

$$\tilde{\tilde{a}}_{uv}^{\langle q\rangle}(k)=\tilde{a}_{rv}^{\langle q\rangle}(k) \text{ at } r=u+s_{\text{ver}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)$$

$$\text{for} \quad u=\overline{-s_{\text{ver}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)+1,\ 60} \text{ and } v=\overline{1,\ 80}. \qquad (16)$$

Clearly, for $s_{\text{ver}}\left(\sigma_{\text{shift}}^{\langle k\rangle}\right)=0$ the q-th horizontally shifted image is not shifted vertically:

$$\tilde{\tilde{a}}_{uv}^{\langle q\rangle}(k)=\tilde{a}_{uv}^{\langle q\rangle}(k) \text{ for } u=\overline{1,\ 60} \text{ and } v=\overline{1,\ 80}. \qquad (17)$$

After all 26 images have become SNMI, each q-th image as matrix $\tilde{\tilde{\mathbf{A}}}_q(k)=\left[\tilde{\tilde{a}}_{uv}^{\langle q\rangle}(k)\right]_{60\times 80}$ is reshaped into 4800-length-column, $q=\overline{1,\ 26}$. Then the matrix $\mathbf{A}_{\text{shift}}^{\langle k\rangle}=\left[\tilde{\tilde{a}}_{jq}(k)\right]_{4800\times 26}$ of all 26 SNMI, reshaped into 26 columns, is included into the training set

$$\tilde{P}_{\text{train}}=\left\{\tilde{\mathbf{P}}_i\right\}_{i=1}^{C+F}=\left\{\left\{\mathbf{A}\right\}_{l=1}^{C},\left\{\mathbf{A}_{\text{shift}}^{\langle k\rangle}\right\}_{k=1}^{F}\right\} \qquad (18)$$

that feeds the input of 2LP, passing through 2LP with identifiers (4) for Q_{pass} times.

Once again, for setting the size of hidden layer of 2LP to 250 neurons at $\sigma_{\text{shift}}^{\langle\max\rangle}=1$, $C=2, F=8$, $Q_{\text{pass}}=10$, and using the training MATLAB-function «traingda», the results of classifying SMI, when 2LP is trained with SNMI and batch-tested, still appear unsatisfactory (figure 4). However, now these results are much better than those ones, derived from 1000 batch testings of the PNMI-trained 2LP in figure 1. But the training process for SNMI is running very lingeringly. Besides, this process may frequently be non-convergent, where some performance goals aren't met, or minimum gradient is reached just after the first pass (in this case 2LP cannot be called the trained with SNMI). The results of the letter-by-letter testing of the SNMI-trained 2LP at some fixed standard deviation for (6) and (7) disclose the peculiar trend in distribution of recognition errors percentage over letters (figure 5), where letters «I» and «L» are the most recognizable, whereas letters «G», «K», «O», «R», «V» are classified wrong in every second case, roughly. The averaged recognition errors percentage, being lower than for the PNMI-trained 2LP, nonetheless remains high.

Having analyzed the performance of the SNMI-lingering-trained 2LP, there is a proposition to shorten the training process by modifying the type of noise. It is verisimilar that adding some pixel noise along with not increasing or lowering the shift intensity may relatively accelerate the training process of 2LP. Also it may decrease the recognition errors percentage. So, the following model is for making pixel-shift-noised monochrome images (PSNMI) to feed the input of 2LP, as neither PNMI-trained 2LP, nor SNMI-trained 2LP is the good classifier of SMI.

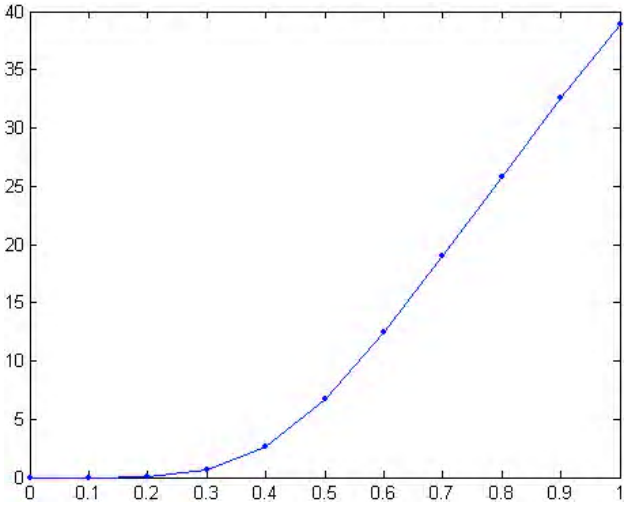

Fig. 4. Percentage of recognition errors p_{error} over standard deviation range $\left[0;\ \sigma_{\text{shift}}^{\langle\max\rangle}\right]$ by 250 neurons in 2LP hidden layer and $\sigma_{\text{shift}}^{\langle\max\rangle}=1$, $\sigma_{\text{pixel}}^{\langle\max\rangle}=0$ (derived from 1000 batch testings of the trained 2LP with SNMI by $\sigma_{\text{shift}}^{\langle\max\rangle}=1$, $C=2$, $F=8$, $Q_{\text{pass}}=10$)

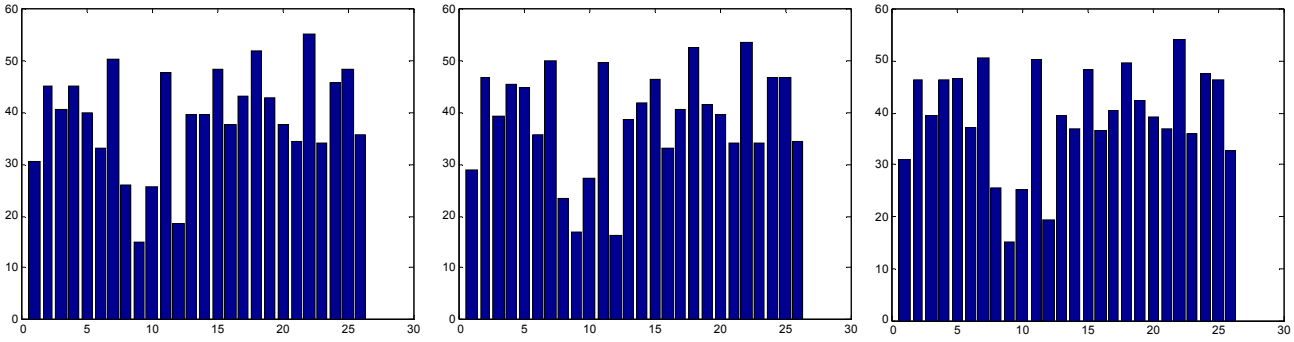

Fig. 5. Distributions of recognition errors percentage over letters at fixed standard deviation $\sigma_{\text{shift}}^{\langle\max\rangle}=1$ for $\sigma_{\text{pixel}}^{\langle\max\rangle}=0$ (derived from 1000 testings of the SNMI-trained-and-loaded 2LP at $\sigma_{\text{shift}}^{\langle\max\rangle}=1$, $C=2$, $F=8$, $Q_{\text{pass}}=10$)

MODEL OF PSNMI

The main principle in modeling PSNMI is that there firstly should be accomplished HPS (6) and VPS (7) for consecutive transformation of matrices $\left\{\mathbf{A}_q\right\}_{q=1}^{26}=\left\{\left[a_{uv}^{\langle q\rangle}\right]_{60\times 80}\right\}_{q=1}^{26}$ through $\left\{\tilde{\mathbf{A}}_q(k)\right\}_{q=1}^{26}=\left\{\left[\tilde{a}_{uv}^{\langle q\rangle}(k)\right]_{60\times 80}\right\}_{q=1}^{26}$ and $\left\{\tilde{\tilde{\mathbf{A}}}_q(k)\right\}_{q=1}^{26}=\left\{\left[\tilde{\tilde{a}}_{uv}^{\langle q\rangle}(k)\right]_{60\times 80}\right\}_{q=1}^{26}$ into the matrix $\mathbf{A}_{\text{shift}}^{\langle k\rangle}=\left[\overline{\overline{\tilde{a}}}_{jq}(k)\right]_{4800\times 26}$ at k-th part of forming the set that feeds the input of 2LP by (5). With the matrix $\mathbf{A}_{\text{shift}}^{\langle k\rangle}$ there follows the model of PNMI (1), that is

$$\mathbf{A}_{\text{pixel-shift}}^{\langle k\rangle}=\mathbf{A}_{\text{shift}}^{\langle k\rangle}+\sigma_{\text{pixel}}^{\langle k\rangle}\cdot\boldsymbol{\Xi} \tag{19}$$

by (2) and (5). For PSNMI the input of 2LP is fed with the training set

$$\tilde{P}_{\text{train}}=\left\{\tilde{\mathbf{P}}_i\right\}_{i=1}^{C+F}=\left\{\left\{\mathbf{A}\right\}_{l=1}^{C},\left\{\mathbf{A}_{\text{pixel-shift}}^{\langle k\rangle}\right\}_{k=1}^{F}\right\} \tag{20}$$

of C replicas of undistorted images and pixel-shift-distorted images by the set of identifiers (4).

Let $\sigma_{\text{shift}}^{\langle\max\rangle}=\sigma_{\text{pixel}}^{\langle\max\rangle}=1$. The training process under such relationship is pretty hard: there are needed many passes, lasting for great numbers of epochs; the weak convergence is very likely, and the hang-up is observed after first 20–30 passes are completed. That means that for covering the bad shift noise (shift noise of high intensity) the standard deviations $\sigma_{\text{shift}}^{\langle\max\rangle}$ and $\sigma_{\text{pixel}}^{\langle\max\rangle}$ mustn't be equal. Truly, here pixel distortion should be either strengthened or loosened. Thus let $\sigma_{\text{shift}}^{\langle\max\rangle}=1=\frac{2}{3}\sigma_{\text{pixel}}^{\langle\max\rangle}$. Unfortunately, under this ratio the training process has the same weak convergence, and its hopeless hang-up is observed after first 20–30 passes are completed. Moreover, if to vary the ratio between $\sigma_{\text{shift}}^{\langle\max\rangle}$ and $\sigma_{\text{pixel}}^{\langle\max\rangle}$ by setting strictly $\sigma_{\text{shift}}^{\langle\max\rangle}=1$ then the training process hangs for any $\sigma_{\text{pixel}}^{\langle\max\rangle}>0{,}5$. Letting $\sigma_{\text{shift}}^{\langle\max\rangle}=1=2\sigma_{\text{pixel}}^{\langle\max\rangle}$, after having been trained with the set (20) for $Q_{\text{pass}}>100$, the PSNMI-trained 2LP produces yet higher performance, which is still unsatisfactory. For $\sigma_{\text{shift}}^{\langle\max\rangle}=1=4\sigma_{\text{pixel}}^{\langle\max\rangle}$ and $Q_{\text{pass}}>150$, the PSNMI-trained 2LP produces yet higher performance than the SNMI-trained 2LP (figure 6) over standard deviation range $\left[0;\sigma_{\text{shift}}^{\langle\max\rangle}\right]=[0;1]$. Testing the PSNMI-trained 2LP for classifying SMI letter-by-letter at the highest noise intensity certifies it (figure 7).

Clearly that the greater Q_{pass} the lower p_{error} over standard deviation range $\left[0;\sigma_{\text{shift}}^{\langle\max\rangle}\right]$ is, although it lingers the training process almost to the long while as it for the SNMI-trained 2LP is (and much longer). However, for the PSNMI-trained 2LP by $\sigma_{\text{shift}}^{\langle\max\rangle}=4\sigma_{\text{pixel}}^{\langle\max\rangle}=1$ almost 30–40 passes performance goals are met, and 2LP can be trained with SNMI with many unmet performance goals. Letting standard deviations $\sigma_{\text{shift}}^{\langle\max\rangle}$ and $\sigma_{\text{pixel}}^{\langle\max\rangle}$ be different from their relationship $\sigma_{\text{shift}}^{\langle\max\rangle}=4\sigma_{\text{pixel}}^{\langle\max\rangle}=1$ may slightly change trends in figures 6 and 7, but upon the whole, the trained 2LP with PSNMI by $Q_{\text{pass}}=234$ is the good classifier of SMI, especially when the shift intensity is defined within standard deviation range $(0;0{,}4]$. The averaged recognition errors percentage doesn't exceed 3,5 %, and this 2LP performs well in classifying shifted monochrome 60-by-80-images at maximal intensity shift noise, when HPS and VPS are about 10–20 pixels and $p_{\text{error}}\approx 15$ with nearly

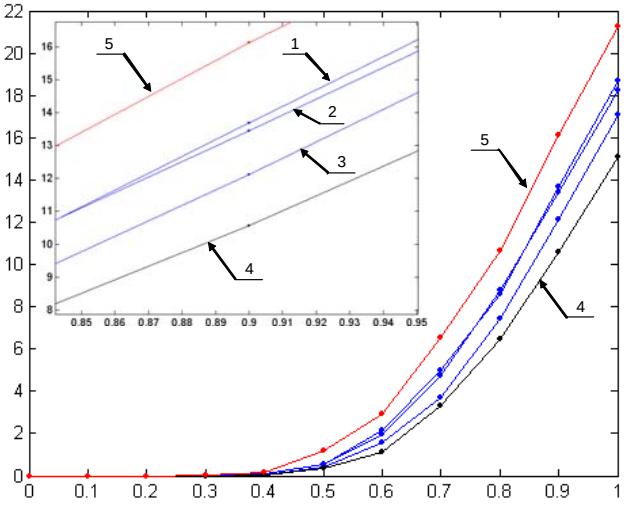

Fig. 6. Percentage of recognition errors p_{error} over standard deviation range $\left[0;\ \sigma_{\text{shift}}^{\langle\max\rangle}\right]$ by 250 neurons in 2LP hidden layer and $\sigma_{\text{shift}}^{\langle\max\rangle}=1,\ \sigma_{\text{pixel}}^{\langle\max\rangle}=0$: 1 – derived from 1000 batch testings of the trained 2LP with PSNMI by $\sigma_{\text{shift}}^{\langle\max\rangle}=4\sigma_{\text{pixel}}^{\langle\max\rangle}=1,\ C=2,\ F=8,$ $Q_{\text{pass}}=175$ (total traintime is 278548 epochs); 2 – derived from 1000 batch testings of the trained 2LP with PSNMI by $\sigma_{\text{shift}}^{\langle\max\rangle}=4\sigma_{\text{pixel}}^{\langle\max\rangle}=1,\ C=2,\ F=8,\ Q_{\text{pass}}=200$ (total traintime is 315134 epochs) 3 – derived from 1000 batch testings of the trained 2LP with PSNMI by $\sigma_{\text{shift}}^{\langle\max\rangle}=4\sigma_{\text{pixel}}^{\langle\max\rangle}=1,\ C=2,\ F=8,\ Q_{\text{pass}}=225$ (total traintime is 329376 epochs); 4 – derived from 1000 batch testings of the trained 2LP with PSNMI by $\sigma_{\text{shift}}^{\langle\max\rangle}=4\sigma_{\text{pixel}}^{\langle\max\rangle}=1,\ C=2,\ F=8,\ Q_{\text{pass}}=234$ (total traintime is 339971 epochs); 5 – derived from 1000 batch testings of the trained 2LP with SNMI by $\sigma_{\text{shift}}^{\langle\max\rangle}=1,\ C=2,$ $F=8,\ Q_{\text{pass}}=100$ (total traintime is 229646 epochs)

every seventh letter is classified wrong (note that letter «I» is the most recognizable, but letter «D» is classified wrong in every fourth case).

CONCLUSION

Problems of classification of SMI are more widespread than problems of classifying PNMI. But the PSNMI-trained 2LP may classify SMI with pixel distortion successfully also, as the training set (20) contains modeled PNMI. The pattern for a monochrome 60-by-80-image, used here, is obviously not general. Nevertheless, the letter is just a model of image, wherewith 2LP can be tested for SMI classifier. And those tests proved that introducing the model of PNMI into the model of SNMI shortens the training process of 2LP and improves its performance in classifying SMI or SMI with pixel distortion. Hence, 2LP is capable to ensure high performance for SMI. There only stays the question of in what ratio standard deviations $\sigma_{\text{shift}}^{\langle\max\rangle}$ and $\sigma_{\text{pixel}}^{\langle\max\rangle}$ should be taken, that is how badly SMI must be distorted to reach the optimal traintime and recognition errors percentage. This question will be examined in further investigations.

SPISOK LITERATURY

1. *Arulampalam, G.* A generalized feedforward neural network architecture for classification and regression / G. Arulampalam, A. Bouzerdoum // Neural Networks. – 2003. – Volume 16, Issues 5–6. – P. 561–568.
2. *Hagan, M. T.* Neural Networks for Control / M. T. Hagan, H. B. Demuth // Proceedings of the 1999 American Control Conference, San Diego, CA. – 1999. – P. 1642–1656.
3. *Benardos, P. G.* Optimizing feedforward artificial neural network architecture / P. G. Benardos, G.-C. Vosniakos // Engineering Applications of Artificial Intelligence. – 2007. – Volume 20, Issue 3. – P. 365–382.
4. *Hagan, M. T.* Training feedforward networks with the Marquardt algorithm / M. T. Hagan, M. B. Menhaj // IEEE Transactions on Neural Networks. – 1994. – Volume 5, Issue 6. – P. 989–993.
5. *Yu, C.* An efficient hidden layer training method for the multilayer perceptron / C. Yu, M. T. Manry, J. Li, P. L. Narasimha // Neurocomputing. – 2006. – Volume 70, Issues 1 – 3. – P. 525–535.

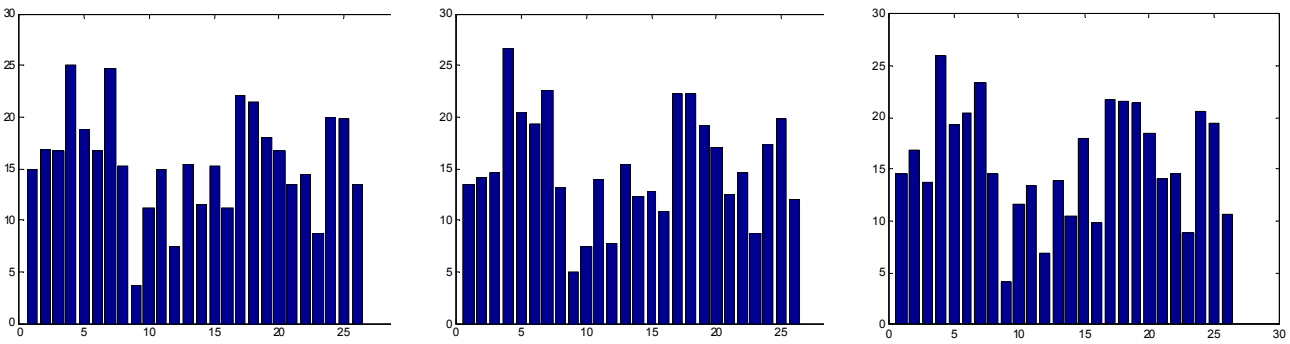

Fig. 7. Distributions of recognition errors percentage over letters at fixed standard deviation $\sigma_{\text{shift}}^{\langle\max\rangle}=1$ for $\sigma_{\text{pixel}}^{\langle\max\rangle}=0$ (derived from 1000 testings of the PSNMI-trained-and-loaded 2LP at $\sigma_{\text{shift}}^{\langle\max\rangle}=4\sigma_{\text{pixel}}^{\langle\max\rangle}=1,\ C=2,\ F=8,\ Q_{\text{pass}}=234$)

6. *Fukushima, K.* Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position / K. Fukushima // Biological Cybernetics. – 1980. – Volume 36, Issue 4. – P. 193–202.
7. *Fukushima, K.* Neocognitron: A hierarchical neural network capable of visual pattern recognition / K. Fukushima // Neural Networks. – 1988. – Volume 1, Issue 2. – P. 119–130.
8. *Hagiwara, K.* Upper bound of the expected training error of neural network regression for a Gaussian noise sequence / K. Hagiwara, T. Hayasaka, N. Toda, S. Usui, K. Kuno // Neural Networks. – 2001. – Volume 14, Issue 10. – P. 1419–1429.
9. *Романюк, В. В.* Зависимость производительности нейросети с прямой связью с одним скрытым слоем нейронов от гладкости ее обучения на зашумленных копиях алфавита образов / В. В. Романюк // Вісник Хмельницького національного університету. Технічні науки. – 2013. – № 1. – С. 201–206.

Стаття надійшла до редакції 10.10.2013.

Романюк В. В.
Канд. техн. наук, доцент, Хмельницкий национальный университет, Украина

ВЫСОКАЯ ПРОИЗВОДИТЕЛЬНОСТЬ ДВУХСЛОЙНОГО ПЕРСЕПТРОНА В КЛАССИФИКАЦИИ МОНОХРОМНЫХ ИЗОБРАЖЕНИЙ ФОРМАТА 60-НА-80 СО СДВИГОМ НА ОСНОВЕ ОБУЧЕНИЯ ПО ИЗОБРАЖЕНИЯМ СО СДВИГОМ И ПИКСЕЛЬНЫМИ ИСКАЖЕНИЯМИ В НАБОРЕ ИЗ 26 АЛФАВИТНЫХ БУКВ

Рассматривается задача объектной классификации, где могут применяться неокогнитрон и многослойный персептрон. Поскольку неокогнитрон, решая практически любую задачу классификации, работает слишком медленно и затратно, то для распознавания монохромных изображений со сдвигом предлагается испробовать двухслойный персептрон, хотя он является быстродействующим только для монохромных изображений с пиксельными искажениями. Предложив набор оригинальных изображений из 26 монохромных изображений букв английского алфавита формата 60-на-80, формулируется задача выяснить, способен ли двухслойный персептрон обеспечить высокую производительность при классификации монохромных изображений со сдвигом. Соответственно обнаруживается, что двухслойный персептрон работает как хороший классификатор монохромных изображений со сдвигом, когда при обучении на его вход поступают обучающие выборки из изображений со сдвигом, пиксели которых искажены. Для этого, однако, может потребоваться больше проходов обучающих выборок через двухслойный персептрон, но тем не менее общее время обучения будет меньше, чем для обучения двухслойного персептрона лишь на монохромных изображениях со сдвигом без пиксельных искажений.

Ключевые слова: объектная классификация, неокогнитрон, персептрон, пиксельное искажение, сдвиг, монохромные изображения, монохромные изображения со сдвигом.

Романюк В. В.
Канд. техн. наук, доцент, Хмельницький національний університет, Україна

ВИСОКА ПРОДУКТИВНІСТЬ ДВОШАРОВОГО ПЕРСЕПТРОНУ В КЛАСИФІКАЦІЇ МОНОХРОМНИХ ЗОБРАЖЕНЬ ФОРМАТУ 60-НА-80 ЗІ ЗСУВОМ НА ОСНОВІ НАВЧАННЯ ПО ЗОБРАЖЕННЯМ ЗІ ЗСУВОМ ТА ПІСКЕЛЬНИМИ СПОТВОРЕННЯМИ У НАБОРІ З 26 АЛФАВІТНИХ ЛІТЕР

Розглядається задача об'єктної класифікації, де можуть застосовуватись неокогнітрон і багатошаровий персептрон. Оскільки неокогнітрон, розв'язуючи практично будь-яку задачу класифікації, працює занадто повільно і затратно, то для розпізнавання монохромних зображень зі зсувом пропонується випробувати двошаровий персептрон, хоча він є швидкодіючим тільки для монохромних зображень з піксельними спотвореннями. Запропонувавши набір оригінальних зображень з 26 монохромних зображень літер англійського алфавіту формату 60-на-80, формулюється задача вияснити, чи здатен двошаровий персептрон забезпечити високу продуктивність при класифікації монохромних зображень зі зсувом. Відповідно виявляється, що двошаровий персептрон працює як хороший класифікатор монохромних зображень зі зсувом, коли при навчанні на його вхід поступають навчальні вибірки зображень зі зсувом, пікселі яких спотворені. Для цього, однак, може знадобитись більше проходів навчальних вибірок через двошаровий персептрон, але тим не менше загальний час навчання буде меншим, ніж для навчання двошарового персептрону лише на монохромних зображеннях зі зсувом без піксельних спотворень.

Ключові слова: об'єктна класифікація, неокогнітрон, персептрон, піксельне спотворення, зсув, монохромні зображення, монохромні зображення зі зсувом.

REFERENCES

1. Arulampalam G., Bouzerdoum A. A generalized feedforward neural network architecture for classification and regression, *Neural Networks*, 2003, Vol. 16, Iss. 5–6, pp. 561–568.
2. Hagan M. T., Demuth H. B. Neural Networks for Control, *Proceedings of the 1999 American Control Conference*, San Diego, CA, 1999, pp. 1642–1656.
3. Benardos P. G., Vosniakos G.-C. Optimizing feedforward artificial neural network architecture, *Engineering Applications of Artificial Intelligence*, 2007, Vol. 20, Iss. 3, pp. 365–382.
4. Hagan M. T., Menhaj M. B. Training feedforward networks with the Marquardt algorithm, *IEEE Transactions on Neural Networks*, 1994, Vol. 5, Iss. 6, pp. 989–993.
5. Yu C., Manry M. T., Li J., Narasimha P. L. An efficient hidden layer training method for the multilayer perceptron, *Neurocomputing*, 2006, Vol. 70, Iss. 1 – 3, pp. 525–535.
6. Fukushima K. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, *Biological Cybernetics*, 1980, Vol. 36, Iss. 4, pp. 193–202.
7. Fukushima K. Neocognitron: A hierarchical neural network capable of visual pattern recognition, *Neural Networks*, 1988, Vol. 1, Iss. 2, pp. 119–130.
8. Hagiwara K., Hayasaka T., Toda N., Usui S., Kuno K. Upper bound of the expected training error of neural network regression for a Gaussian noise sequence, *Neural Networks*, 2001, Vol. 14, Iss, 10, pp. 1419–1429.
9. Romanuke V. V. Dependence of performance of feed-forward neuronet with single hidden layer of neurons against its training smoothness on noised replicas of pattern alphabet, *Bulletin of Khmelnitskiy National University. Technical Sciences*, 2013, No. 1, pp. 201–206.