UDC 004.032.26::004.93+519.8

Romanuke V. V.

Dr. Sc., Professor, Professor of the Department of Applied Mathematics and social Informatics, Khmelnitskiy National University, Ukraine

A FRAMEWORK FOR CLASSIFIER SINGLE TRAINING PARAMETER OPTIMIZATION ON TRAINING TWO-LAYER PERCEPTRON IN A PROBLEM OF TURNED 60-BY-80-IMAGES CLASSIFICATION

A 13-itemed scenario framework for classifier single training parameter optimization is developed. Formally, the problem is to find global extremum (mostly, minimum) of function as a classifier output parameter against its single training parameter. Linking the scenario theory to praxis, the classifier type has been decided on two-layer perceptron. Its input objects are monochrome images of a medium format, having a few thousands independent features. Within the framework, the programming environment has been decided on MATLAB, having powerful Neural Network Toolbox. Keeping in mind the stochasticity of the being minimized function, there is defined statistical ε-stability of its evaluation by a finite set of data. These data are mined in batch testings of the trained classifier. For exemplification of the scenario framework, there is optimized pixel-to-turn standard deviations ratio for training two-layer perceptron in classifying monochrome 60-by-80-images of the enlarged 26 English alphabet capital letters. The goal is to find a pixel-to-turn standard deviations ratio for the training process in order to ensure minimum of classification error percentage. The optimization relative gain is about a third. The developed framework can be applied also for classifier multivariable optimization, wherein it instructs which item operations shall regard the corresponding multiplicity of variables.

Keywords: classifier training parameter optimization, statistical evaluation, optimization scenario, two-layer perceptron, classification error percentage, turned objects classification, monochrome image, pixel-to-turn standard deviations ratio, training set.

NOMENCLATURE

CEP is a classification error percentage;

CNN is a convolutional neural network;

COS is a classifier operation speed;

GT is a general totality;

HMLNN is a hierarchical multilayered neural network;

IMR is an information-memory resources;

IRGSS is an independently random-generated subsets;

LIF is a lot of independent features;

4800-LCRM is a 4800-length-column-reshaped matrices;

2LP is a two-layer perceptron;

MI is a monochrome image;

MOZ is a matrices of ones and zeros;

PD is a pixel distortion;

PDTM6080I is a pixel-distorted TM6080I;

PTSDR is a pixel-to-turn standard deviations ratio;

4800-250-26-P 2LP is a having 4800 neurons in the input layer, 250 in SHL, and 26 in the output;

SD is a standard deviation;

SHL is a single hidden layer;

NDR is a non-distorted representatives;

NV is a normal variate;

TD is a turn distortion;

TM6080I is a turned monochrome 60-by-80-image;

ZEUV is a zero expectation and unit variance;

ε-SSE is a statistically ε-stable evaluation;

$\mathbf{A}_q$ is a matrix representing the q-th class NDR;

$\widehat{\mathbf{A}}_q(k)$ is a 60×80 matrix as the q-th class representative TM6080I formed at the k-th stage;

$\mathbf{A}^{\langle k \rangle}_{\mathrm{PDTM6080I}}$ is a 4800×26 matrix of PDTM6080I of all NDR formed at the k-th stage;

$\mathbf{A}^{\langle k \rangle}_{\mathrm{TM6080I}}$ is a 4800×26 matrix of TM6080I of all NDR formed at the k-th stage;

C is a number of pure objects' replicas;

F is a number indicating at smoothness in training the perceptron;

$f(\alpha)$ the classifier output parameter influenced with the training parameter α;

G is a set of indices;

$\mathbf{I}$ is a identity matrix;

n is a number of the sampled points from the segment of the parameter values;

N_{classes} is a number of classes;

N_{features} is a number of the object features;

$p_{\mathrm{EC}}(r)$ is an averaged CEP;

Q_{pass} is a number of training samples to be passed through 2LP;

r is a variable of PTSDR;

$\mathbf{T}_i$ is a target matrix for the i-th training subsample;

α is a training parameter influencing significantly on the classifier output parameter;

$\beta(k)$ the angle through which NDR is turned;

© Romanuke V. V., 2014

DOI 10.15588/1607-3274-2014-2-13

σ_{PD} is a SD of PD;

σ_{TD} is a SD of TD;

$\sigma_{\text{PD}}^{\langle k \rangle}$ is a SD of PD at the k-th stage of PDTM6080I set formation;

$\sigma_{\text{TD}}^{\langle k \rangle}$ is a SD of TD at the k-th stage of TM6080I set formation;

$\sigma_{\text{PD}}^{\langle \max \rangle}$ is a maximum of SD in PD;

$\sigma_{\text{TD}}^{\langle \max \rangle}$ is a maximum of SD in TD;

$\mathbf{\Xi}$ is a matrix of values of NV with ZEUV;

$\xi(k)$ is a value of NV with ZEUV, raffled at the k-th stage of TM6080I set formation.

INTRODUCTION

The object classification is an important constituent in automation processes [1, 2]. Main hardness in classifying objects arises when the object has LIF, and anyone of objects cannot be ignored for simplification. Usually, a LIF object is a high resolution image after preprocessing [2, 3]. Such images are used in the manufacturing control, technical and medical diagnostics, safety watching systems, astrophysical explorations, and so forth [1, 3, 4]. A hard problem of LIF objects classification lies in COS. More particularly, COS is bettered by reducing the huge IMR which are consumed by the classification system, including the classifier [2, 4, 5]. The classifiers on the basis of HMLNN, CNN, cognitrons and neocognitrons perform perfectly over objects for their classification. At the same time, such classifiers take long periods for the performance and consume gigantic IMR [4, 6, 7]. And vice versa, perceptrons with one or more hidden layers consume far less IMR, while they are much faster. In this way, the perceptron with SHL is frequently used. This is 2LP, and it is the fastest for the most of classification problems, where the object features are distorted by a statistical law having near-normal distribution. However, 2LP cope poorly with classification problems, where object distortions are specific [5, 8]. For images, those specific distortions are skewness, nonlinear scaling, turn, severe rotation, shift, etc. Hence, there is a known duality in the problem of classifying LIF objects. Either we are to better COS and IMR consumption for perfectly performing HMLNN, CNN, cognitrons and neocognitrons, or to optimize some parameters of 2LP to get it trained suitably on specifically distorted LIF objects. And it is plain that the way with 2LP is preferable to bettering COS and IMR consumption with far beyond complicated neuronets.

The article goal is to develop the framework of the scenario type for 2LP classifier single training parameter optimization. At that, the classifier input is fed with LIF objects, making heuristic methods of optimization inapplicable. For achieving the goal, both theoretical and practical phases are to be delineated. Preceding the praxis, theoretical phase is tasked to give a justified succession of jobs whose accomplishment guarantees the classifier influential improvement on its output parameter. The tasking will be replenished by verifying practically whether that output parameter is bettered over the specifically distorted objects. And the conclusive task is to infer from the development and investigation results.

1 PROBLEM STATEMENT

Let, without loss of generality, we have α and $f(\alpha)$, which are both positive. Formally, the decided problem is to find

$$\alpha^{*} \in \arg \operatorname*{extr}_{\alpha > 0} f(\alpha) \tag{1}$$

and get 2LP, performing at $f(\alpha^{*})$. In practicing, classifiers or classification processes mostly have characteristics desired for their minimization. And, naturally, the training parameter α has its bounds $\alpha_{\min}$ and $\alpha_{\max}$, that is $\alpha \in [\alpha_{\min}; \alpha_{\max}]$ for $\alpha_{\max} > \alpha_{\min} > 0$. Henceforth, the problem (1) is rendered concrete:

$$\alpha^{*} \in \arg \min_{\alpha \in [\alpha_{\min};\, \alpha_{\max}]} f(\alpha). \tag{2}$$

2 REVIEW OF THE LITERATURE

2LP in preference to far beyond complicated neural networks has a great many of its parameters. They are divided into the couple of associated parametric groups. The first one is assembled from size of SHL and transfer functions [5]. The second one is assembled from parameters of the training algorithm, methods of the algorithm implementation, and methods of forming training samples [8]. All these parameters and methods are formalized as training parameters [9, 10].

Typically, SHL size and transfer functions of 2LP are set up heuristically, being based on sizes of input and output layers and recent experience [5, 8, 9, 11]. The classifier training parameters optimization is only possible before the classifier is trained. Once 2LP has been trained, the classification process attains its characteristics. Stable estimations of these characteristics can be valued after huge batch testing.

The training process is defined with the set of training samples, feeding the input of 2LP [5, 8, 12]. Parameters of the set of training samples in the training process are specific for a classification problem [5]. To get the classification process improved, training samples are modified accordingly to types of object distortion [10, 13]. The modification has its own parameter or parameters. Sweeping them gives, speaking factually, diverse classifiers. And the best classifier could be chosen among those ones.

The best classifier choice nonetheless requires a framework of successive operations over the classifier training parameters to get them optimized. Before getting started, a parameter to be swept is to be appointed along with the classification process output parameter, which is influenced with the being swept parameter. Hence there is a want of developing the scenario of making the classifier optimization upon its parameter from a raw 2LP up to the classifier having one of its characteristics improved.

3 MATERIALS AND METHODS

For solving the problem (1), the function $f(\alpha)$ evaluation requires defining the object type. The object type is actually defined with its format containing information about the number N_{features}. Dimensions do not matter because the object initial mathematical presentation (aggregate of its features) is reshaped into line array.

A convenient object type is MI of a medium format, having a few thousands features. This type and format allows watching and rendering MI distortions readily. The medium format ensures delayless investigation procedures and the classification results acquisition. And those results are going to be powerfully propagated on other objects of medium formats.

Within MATLAB, having powerful Neural Network Toolbox [14, 15] for programming and simulating neural networks, MI bitmap file is coded with ones (white color) and zeros (black color). Therefore the finite subset of GT is of MOZ, containing altogether $2^{N_{\text{features}}}$ MI. Wholly, GT is infinite, wherein the pixel's value is not necessarily equal to 1 or 0, but it is a real number. Namely within the infinite GT, the specifically distorted objects must be recognized and classified.

After the object type is defined, number $N_{\text{classes}} \in \mathbb{N} \setminus \{1\}$ is prescribed. And so there is defined GT containing all permissible objects of the defined type. Within the defined GT, NDR of classes are fixed.

Having the object type, N_{classes} NDR and GT, the number of neurons in SHL of 2LP is assigned. Rules for SHL neurons number assignation are drilled on experience [5, 11, 16]. Assigning transfer functions in the layers of 2LP is the last item in the configuration of 2LP.

Having had the configuration of 2LP, there is a MATLAB function for training to be assigned. Then the configured 2LP is initialized on MATLAB Neural Network Toolbox simply with the function «feedforwardnet» (the similar function «newff» if for MATLAB versions before R2010).

The function $f(\alpha)$ evaluation starts with estimating the endpoints of the range $[\alpha_{\min}; \alpha_{\max}]$. Having estimated them, this range is to be sampled with a step. This gives a finite subset of points of the training parameter α. 2LP is trained over this subset, and so the statistics of the trained 2LP is accumulated. By the way, the sampling step can be non-constant if needed, so the subset is

$$\{\alpha_h\}_{h=1}^{n} \subset [\alpha_{\min}; \alpha_{\max}] \text{ by } n \in \mathbb{N} \setminus \{1, 2\}$$

$$\text{at } \alpha_1 = \alpha_{\min} \text{ and } \alpha_n = \alpha_{\max}. \tag{3}$$

Really, after the sampling (3) the problem (2) is

$$\alpha^* \in \arg \min_{\alpha \in \{\alpha_h\}_{h=1}^{n}} f(\alpha) \tag{4}$$

with the function evaluation

$$f(\alpha) = \frac{1}{|G|} \sum_{g \in G} f^{\langle g \rangle}(\alpha) \tag{5}$$

by the g-th polyline $f^{\langle g \rangle}(\alpha)$, whose values $\left\{ f^{\langle g \rangle}(\alpha) \right\}_{\alpha \in \{\alpha_h\}_{h=1}^{n}}$ are taken via testing the trained 2LP in each of n points $\{\alpha_h\}_{h=1}^{n}$, repeated for $|G|$ times. Speaking strictly, the batch testings are sufficient for evaluating the function $f(\alpha)$ adequately if for some $\varepsilon > 0$

$$\left| \tilde{f}_1(\alpha) - \tilde{f}_2(\alpha) \right| < \varepsilon \quad \forall\, \alpha \in [\alpha_{\min}; \alpha_{\max}] \tag{6}$$

by

$$\tilde{f}_1(\alpha) = \frac{1}{|G_1|} \sum_{g \in G_1 \subset G} f^{\langle g \rangle}(\alpha),$$

$$\tilde{f}_2(\alpha) = \frac{1}{|G_2|} \sum_{g \in G_2 \subset G} f^{\langle g \rangle}(\alpha) \tag{7}$$

for the probability $P(G_1 \cap G_2 \neq \varnothing)$ which is such that

$$\lim_{|G| \to \infty} P(G_1 \cap G_2 \neq \varnothing) = 0. \tag{8}$$

For wide ranges and insufficiently completed set of indices G, having its insufficiently great cardinal $|G|$, the condition (6) fails. However, the segment $[\alpha_{\min}; \alpha_{\max}]$ may be narrowed to a subsegment by the following reason. If there is great likelihood of that

$$f(\alpha_1) \leqslant f(\alpha_2) \text{ for } \alpha_{\min} < \alpha_1 < \alpha_2 < \alpha_{\max} \tag{9}$$

then we can narrow the segment $[\alpha_{\min}; \alpha_{\max}]$ to $[\alpha_{\min}; \alpha_0]$ by $\alpha_0 \in (\alpha_1; \alpha_2)$ or, for instance, to $[\alpha_{\min}; (\alpha_1 + \alpha_2)/2]$. Then we proceed in batch testings with re-estimation of the right endpoint $\alpha_{\max} = \alpha_0$. Statistical sufficiency is very important here. Thence is an unambiguous definition concerning sufficiency.

Definition 1. Function (5) is called ε-SSE by the values

$$\left\{ \left\{ f^{\langle g \rangle}(\alpha) \right\}_{\alpha \in \{\alpha_h\}_{h=1}^{n}} \right\}_{g \in G} \tag{10}$$

if for fixed $\varepsilon > 0$ and $c(\varepsilon) \in \mathbb{N}$ and any pair $\{G_1, G_2\}$ of IRGSS $\{G_1 \subset G, G_2 \subset G\}$ of the indices' set G with

$$\min\{|G_1|, |G_2|\} \geqslant c(\varepsilon) \tag{11}$$

the inequality

$$\left|\tilde{f}_1(\alpha)-\tilde{f}_2(\alpha)\right|<\varepsilon \quad \forall\ \alpha\in\left\{\alpha_h\right\}_{h=1}^{n} \tag{12}$$

holds by (7) and (8).

So then for approaching to the problem (4) solution, in accordance with Definition 1 we need ε-SSE (5) for sufficiently small ε and great $c(\varepsilon)$. Practically, ε can be assigned to

$$\varepsilon\leqslant\varepsilon_0\cdot\left(\min_{\alpha\in\left\{\alpha_h\right\}_{h=1}^{n}} f(\alpha)\right) \tag{13}$$

by, say, $\varepsilon_0\in\{0.1,\ 0.05,\ 0.02,\ 0.01\}$ (traditional exemplification). The integer $c(\varepsilon)$ is of the order of hundreds at least. And there predictably thrusts the concept of continuity in ε-SSE.

Definition 2.1. Function (5) is called continuous ε-SSE by the values (10) if $\forall\ \varepsilon>0\ \ \exists\ \delta>0$ such that the inequality (12) holds by (7) and

$$P\left(G_1\cap G_2\neq\varnothing\right)<\delta \tag{14}$$

for IRGSS $\{G_1\subset G,\ G_2\subset G\}$ by (8).

An alternative definition to Definition 2.1 handles the relationship between cardinals of IRGSS and G. This one is handy standing off the probability (14) calculation.

Definition 2.2. Function (5) is called continuous ε-SSE by the values (10) if $\forall\ \varepsilon>0\ \ \exists\ \delta\in(0;\ 1]$ such that the inequality (12) holds by (7) and

$$P\left(\frac{\left|G_1\cap G_2\right|}{|G|}<\delta\right)=1 \tag{15}$$

for IRGSS $\{G_1\subset G,\ G_2\subset G\}$ by (8).

These two versions of continuity in ε-SSE are purely theoretical. However, they drive to solving the problem (4) statistically inasmuch as ε-SSE does not determine the minimum point α^* itself. Note that probability (15) is apparently substituted with the expression $\frac{\left|G_1\cap G_2\right|}{|G|}<\delta$.

Definition 3. The point α^* is called solution of the problem (4) with ε-SSE (5) by the values (10) if the statement

$$\alpha^*\in\left\{\arg\min_{\alpha\in\left\{\alpha_h\right\}_{h=1}^{n}}\tilde{f}_1(\alpha)\right\}\cap\left\{\arg\min_{\alpha\in\left\{\alpha_h\right\}_{h=1}^{n}}\tilde{f}_2(\alpha)\right\}\neq\varnothing \tag{16}$$

holds at (7) by any pair of IRGSS $\{G_1\subset G,\ G_2\subset G\}$ with the condition (11) for their cardinals.

Verification of that the training parameter α has been bettered over the specifically distorted objects, having become α^*, is carried out to get convinced of that

$$f\left(\alpha^*\right)<f(\alpha)\quad\forall\ \alpha\in\left\{\alpha_h\right\}_{h=1}^{n}\setminus\left\{\alpha^*\right\}. \tag{17}$$

With the set G for verification, the function evaluation (5) is suitable for (17).

Entirely, scenario for the single training parameter optimization of 2LP classifier is given in 13 items:

1. Defining the object type.
2. Prescribing the number of classes $N_{\text{classes}}\in\mathbb{N}\setminus\{1\}$.
3. Defining GT.
4. Fixing N_{classes} NDR of classes within the defined GT.
5. Assigning SHL neurons number.
6. Assigning transfer functions in the layers of 2LP.
7. Assigning a MATLAB function for training.
8. Appointing the parameter α along with the classification process output parameter $f(\alpha)$.
9. Estimating the range $[\alpha_{\min};\ \alpha_{\max}]$ endpoints.
10. Sampling the range $[\alpha_{\min};\ \alpha_{\max}]$ into the finite subset (3).
11. Proceeding in batch testings of the trained 2LP in each of those points of the finite subset (3) in order to obtain statistically stable evaluation (5) for solving the problem (4).
12. If for fixed ε and $c(\varepsilon)$ the $(n-1)$-segmented polyline (5) is ε-SSE then go to solving the problem (4) and return the result $\left\{\alpha^*,\ f\left(\alpha^*\right)\right\}$ along with verification of that.
13. If the problem (4) has been not solved then go to narrowing the segment $[\alpha_{\min};\ \alpha_{\max}]$ to $[\alpha_{\min};\ \alpha_0]$ by $\alpha_0\in(\alpha_1;\ \alpha_2)$ if possible and recur to the item 10 with $\alpha_{\max}=\alpha_0$; if the narrowing is impossible then recur to the item 11 with the increased $|G|$.

4 EXPERIMENTS

Let's exemplify the developed scenario framework in the problem of turned objects classification. This is a widespread problem, when recognizing plane images or their contours the object rotation effect springs up. It is caused with that there cannot be ensured the symmetry just on the moment when the object is captured after it has been traced.

May the format of MI be 60×80 and may an MI model be the enlarged English alphabet capital letter by $N_{\text{classes}}=26$. Therefore the finite subset of GT is of 60×80 MOZ, containing altogether 2^{4800} MI by $N_{\text{features}}=4800$. Wholly, GT is of matrices whose elements are real numbers. Thus there are 26 capital letters as NDR.

The input layer in 2LP has 4800 neurons, and the output layer has 26 neurons. The size of SHL in 2LP for the problem of classifying TM6080I can be assigned to 250 neurons. Transfer functions are log-sigmoid [5, 8, 10, 12]. This 4800-250-26-P is initialized with «feedforwardnet» («newff») by 1206776 weight and bias values. For training perceptrons there is the backpropagation algorithm, having many methods of its implementation in MATLAB. One of them is the method of a MATLAB function «traingda» [16], where weight and bias values are updated according to gradient descent with adaptive learning rate [10, 12, 17, 18]. Every 4800-250-26-P shall be trained with the function «traingda», guaranteeing quick training process passage.

In the training process for classifying TM6080I, the training set is formed as addition of matrix of pixel-distorted images and matrix of turned images. Intensity (deepness) of PD is measured via SD σ_{PD} and intensity of TD is measured via SD σ_{PD}, determining the range of left and right angulation due to TD. The parameter here is PTSDR $r = \sigma_{\text{PD}} / \sigma_{\text{TD}}$, and the classification process output parameter $f(\alpha)$ is CEP. Hence the problem (2) is

$$r^{*} \in \arg \min_{r \in [r_{\min};\, r_{\max}]} p_{\text{EC}}(r) \tag{18}$$

by the function $p_{\text{EC}}(r)$, whose value at PTSDR r is an averaged CEP over TM6080I or, probably, over PDTM6080I.

PTSDR is constant while training. In the statement $r = \sigma_{\text{PD}} / \sigma_{\text{TD}}$ both SD may vary, so the assignment $r = \sigma_{\text{PD}}^{\langle \max \rangle} \big/ \sigma_{\text{TD}}^{\langle \max \rangle}$ is formally more correct. Due to a model of TM6080I, an NDR is turned through the angle $\beta(k) = (180/\pi) \cdot \sigma_{\text{TD}}^{\langle k \rangle} \xi(k)$ in degrees around the center point of MI, $k = \overline{1,\, F}$ and $F \in \mathbb{N}$. SD in the angle $\beta(k)$ is $\sigma_{\text{TD}}^{\langle k \rangle} = \sigma_{\text{TD}}^{\langle \max \rangle} \cdot k \big/ F \ \forall\ k = \overline{1,\, F}$. From the statement for the angle $\beta(k)$ by SD $\sigma_{\text{TD}}^{\langle k \rangle}$ it follows that the upper value $\sigma_{\text{TD}}^{\langle \max \rangle} = 0.2$ is enough. Beyond this value MI becomes turned so that its turning is impracticable for 2LP classifier, when rotation angle is about $\pi/8$ and severer.

In estimating the range $[r_{\min};\, r_{\max}]$ endpoints, note that on $\sigma_{\text{PD}}^{\langle \max \rangle} > 2$ PDTM6080I becomes over-distorted, and on $\sigma_{\text{PD}}^{\langle \max \rangle} < 0.005$ PD is imperceptible. Then $r_{\min} = 0.025$ and $r_{\max} = 10$ enclose the range $[0.025;\ 10]$ which is to be sampled.

In the training process the input of 4800-250-26-P is fed with the training set

$$\left\{ \mathbf{P}_i^{\langle \text{PDTM6080I} \rangle} \right\}_{i=1}^{C+F} = \left\{ \{\mathbf{A}\}_{l=1}^{C},\ \left\{ \mathbf{A}_{\text{PDTM6080I}}^{\langle k \rangle} \right\}_{k=1}^{F} \right\} \tag{19}$$

of $C \in \mathbb{N}$ replicas of all 26 classes NDR and F matrices of PDTM6080I by targets $\{\mathbf{T}_i\}_{i=1}^{C+F} = \{\mathbf{I}\}_{i=1}^{C+F}$, where 4800×26 matrix $\mathbf{A}$ is formed by concatenating horizontally 4800-LCRM $\{\mathbf{A}_q\}_{q=1}^{26}$ by the q-th class NDR as 60×80 MOZ $\mathbf{A}_q = \left(a_{uv}^{\langle q \rangle} \right)_{60 \times 80}$. At the k-th stage of PDTM6080I set formation, 4800×26 matrix

$$\mathbf{A}_{\text{PDTM6080I}}^{\langle k \rangle} = \mathbf{A}_{\text{TM6080I}}^{\langle k \rangle} + \sigma_{\text{PD}}^{\langle k \rangle} \cdot \mathbf{\Xi} \tag{20}$$

by $\sigma_{\text{PD}}^{\langle k \rangle} = \sigma_{\text{PD}}^{\langle \max \rangle} \cdot k \big/ F \ \forall\ k = \overline{1,\, F}$. Concurrently for (20), 4800×26 matrix $\mathbf{A}_{\text{TM6080I}}^{\langle k \rangle} = \left[\bar{\bar{a}}_{jq}(k) \right]_{4800 \times 26}$ is formed by concatenating horizontally 4800-LCRM $\left\{ \widehat{\mathbf{A}}_q(k) = \left[\widehat{a}_{uv}^{\langle q \rangle}(k) \right]_{60 \times 80} \right\}_{q=1}^{26}$. The q-th class NDR $\mathbf{A}_q$ becomes TM6080I

$$\widehat{\mathbf{A}}_q(k) = 1 - \boldsymbol{\rho}\left(1 - \mathbf{A}_q,\ \beta(k),\ M,\ S \right) \tag{21}$$

with MATLAB function «imrotate» in the map $\boldsymbol{\rho}$, turning the input MI negative $1 - \mathbf{A}_q$ through the angle $\beta(k)$, by the interpolation method handle M and the handle S for specifying the size of the returned negative MI. MI is turned in counterclockwise direction if $\beta(k) > 0$, and for $\beta(k) < 0$ MI is turned clockwise; for $\beta(k) = 0$ MI remains NDR. After all 26 classes NDR have been turned and become TM6080I, the training set (19), formed with (20) by $\sigma_{\text{PD}}^{\langle k \rangle}$ and (21), feeds the input of 4800-250-26-P, passing through 4800-250-26-P with targets $\{\mathbf{T}_i\}_{i=1}^{C+F} = \{\mathbf{I}\}_{i=1}^{C+F}$ for $Q_{\text{pass}} \in \mathbb{N}$ times.

5 RESULTS

In the initial probe, PDTM6080I-trained 4800-250-26-P under set $\{C,\ F,\ Q_{\text{pass}}\} = \{2,\ 8,\ 10\}$ for the training set (19) is tested with TM6080I at $\sigma_{\text{TD}} \in [0;\ 0.2]$. As it has been expected (figure 1), the first approximation over the subset

$$\left\{ 0.025,\ 0.05,\ \{0{,}1h\}_{h=1}^{20},\ \{2 + h\}_{h=1}^{8} \right\} \subset [0.025;\ 10] \tag{22}$$

would not have solved the problem (18)

$$r^{*} \in \arg \min_{r \in [0{,}025;\ 10]} p_{\text{EC}}(r) \tag{23}$$

because the rough 29-segmented polyline of the function $p_{\text{EC}}(r)$ is not ε-SSE. Notwithstanding that PDTM6080I-trained 4800-250-26-P were tested on only TM6080I and $|G| = 40$ appeared to have been underrated, figure 1 prompts that the range $[0.025;\ 10]$ could be narrowed to the subsegment $[0.025;\ 1]$, where $r^{*} \in [0.025;\ 1]$ apparently.

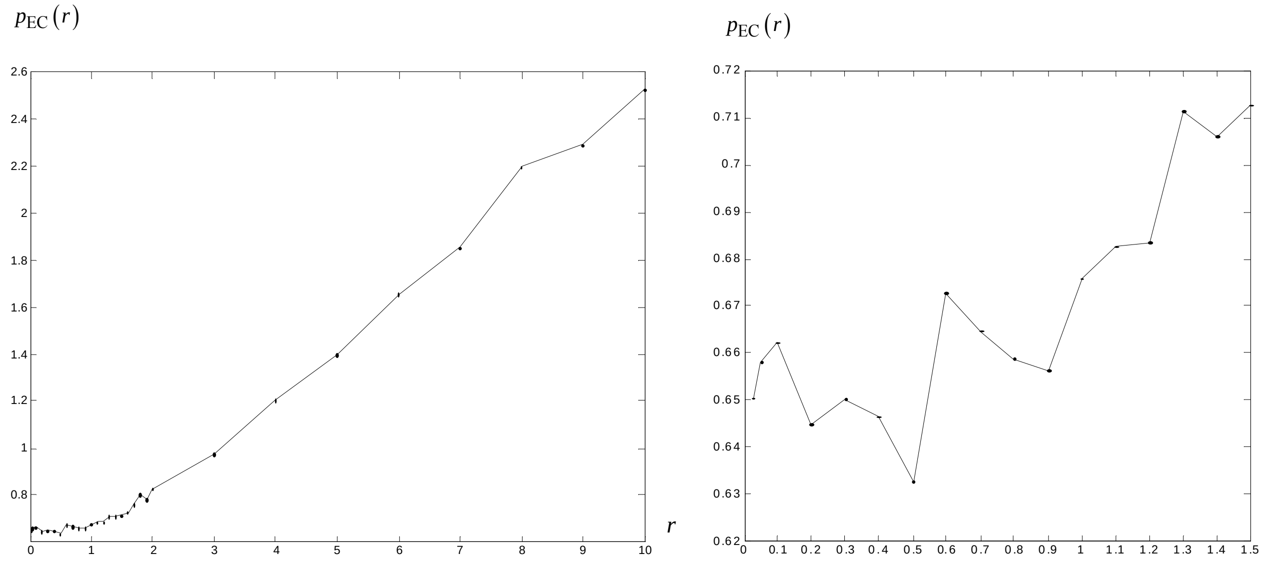

Figure 1 – A rough 29-segmented polyline of the function $p_{\text{EC}}(r)$ over the subset (22), derived from 40 series of 400 batch testings of PDTM6080I-trained 4800-250-26-P (tested on only TM6080I for faster preliminary results)

The second (40 series of 800 batch testings) and third (100 series of 800 batch testings) approximations over the subsets $\{0.025h\}_{h=1}^{40} \subset [0.025;\ 1]$ and $\{0.025h\}_{h=1}^{24} \subset [0.025;\ 0.6]$ correspondingly do not give ε-SSE for the said $|G| = 40$ and $|G| = 100$. Nevertheless, this 0.025-sampling lets feel and see that $r^* \in [0.025;\ 0.6]$, needing though to increase the cardinality of G. Thanking to that all those 140 polylines of the function $p_{\text{EC}}(r)$ were obtained on 0.025-sampling, the cardinality here could be increased just by superposing those 40 and 100 polylines.

However, neither superposition nor another 200 series of 200 batch testings (appeared to be sufficient to understand classification properties of an PDTM6080I-trained 4800-250-26-P) give ε-SSE of the function $p_{\text{EC}}(r)$. Here, although the point $r = 0.475$ could have been ventured to accept it as the problem (23) solution, it as well could be refuted.

Herewith, for obtaining ε-SSE of the function $p_{\text{EC}}(r)$ at some reasonable ε there is $c(\varepsilon) > 340$. And the shape of the 23-segmented polyline of the function $p_{\text{EC}}(r)$ becomes more stochastic when extracting a lesser cardinality subset from those 340 polylines. However, such deep stochasticity may be provoked with the current conformation of training and testing. Likely, number of passes $Q_{\text{pass}} = 10$ is too low for expecting stable estimation. Also PTSDR $r = 5$ for testing is too high, randomizing the output results of estimations additionally.

Nevertheless, there are evidences of that the problem (23) solution is in the segment $[0.025;\ 0.5]$ of PTSDR. The further probe with taking $Q_{\text{pass}} = 25$ and testing PDTM6080I-trained 4800-250-26-P with PDTM6080I on $\sigma_{\text{TD}} \in [0;\ 0.2]$ and $\sigma_{\text{PD}} \in [0;\ 0.2]$ or $\sigma_{\text{PD}} \in [0;\ 0.4]$ returns much the same stochasticity. A corollary is that the upper value of PTSDR for testing does not influence much on stochasticity of $p_{\text{EC}}(r)$.

Final evaluation on $|G| = 370$ (figure 2) shows that the problem (23) solution is $r^* = 0.225$, having been taken off Definition 3, though. In this situation, the most appropriate conclusion would be that $r^* \in (0.075;\ 0.3)$, where the cavity of the polyline in figure 2 by $r \in (0.075;\ 0.3)$ can be seen. And the local minimum at $r = 0.075$ gives almost the same value of CEP therewith.

Henceforward, the best PDTM6080I-trained 4800-250-26-P by

$$\sigma_{\text{PD}}^{\langle\max\rangle} = r^* \cdot \sigma_{\text{TD}}^{\langle\max\rangle} = 0.225 \cdot 0.2 = 0.045 \qquad (24)$$

under the set $\{C,\ F,\ Q_{\text{pass}}\} = \{2,\ 8,\ 25\}$ classifies TM6080I at CEP $p_{\text{EC}}(0.225) < 0.28$ and classifies PDTM6080I at CEP $p_{\text{EC}}(0.225) < 0.48$. Moreover, the best PDTM6080I-trained 4800-250-26-P by (24) under the set $\{C,\ F,\ Q_{\text{pass}}\} = \{2,\ 8,\ 75\}$ classifies TM6080I at CEP $p_{\text{EC}}(0.225) < 0.19$ and classifies PDTM6080I at CEP $p_{\text{EC}}(0.225) < 0.29$, and the averaged CEP is $p_{\text{EC}}(0.225) < 0.24$. As the example visualization, figure 3 shows TM6080I which became recognizable for the trained 2LP classifier after its single training parameter optimization with (24).

The gain of the executed optimization depends on primary heuristics in adjusting PTSDR. Thus, if it were $r = 1$, what could have been mostly expected in adjusting off optimization, then the absolute gain is greater than 0.1 %. In relative calculus, it is about a third. For cases with $r > 1$ the gain is even higher.

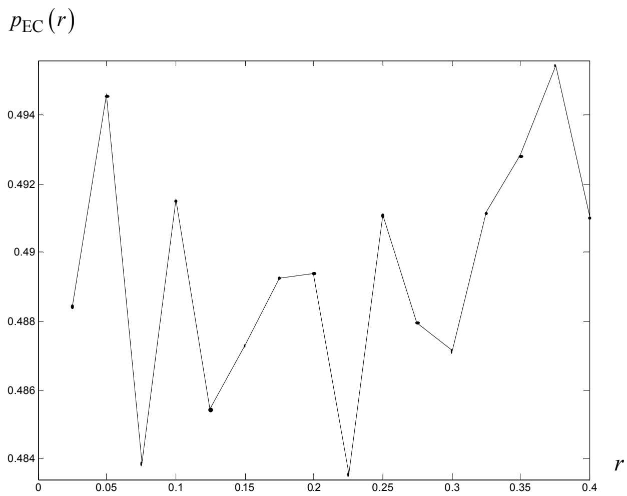

Figure 2 – $0,01$-SSE of the function $p_{\mathrm{EC}}(r)$ on $\sigma_{\mathrm{TD}} \in [0;\ 0,2]$ and $\sigma_{\mathrm{PD}} \in [0;\ 0,4]$, derived from 370 series of 400 batch testings of PDTM6080I-trained 4800-250-26-P

Eventually, the 13-itemed scenario framework for classifier single training parameter optimization allowed to have trained 4800-250-26-P on PDTM6080I optimally in order to perform classification TM6080I at the lowest CEP. And PDTM6080I are classified well also. The solution (23) can be applied for training 2LP classifiers in other problems of turned LIF objects classification, where number of features is about 4800 and number of classes is about 26.

6 DISCUSSION

The 13-itemed scenario framework suggests a straightforward succession of operations to get 2LP classifier single training parameter optimized. Furthermore, this framework with proper modifications (items 5, 6, and 7) is applicable for optimizing training parameters of HMLNN and CNN. Besides, the classifier input is not necessary to be fed with LIF objects. The gain is either the lowest CEP or getting a classifier output parameter bettered.

The operations from item 1 through item 8 do not require statistics. Otherwise, for narrowing the segment $[\alpha_{\min};\ \alpha_{\max}]$ we need great likelihood of that (9) holds. The greatness is ascertained via series of the trained classifier's batch testings, where the series number is of the order of several tens (hundreds) at least.

The hardest operation is the item 12. For ascertaining statistical ε-stability of the polyline (5), the requirement (13) is purely practical. It is clear that for implementing the conditions of Definition 1, the integer $c(\varepsilon)$ is specified by experience for a few tens of IRGSS $\{G_1,\ G_2\}$. These subsets can be used also in Definition 2.2, before setting up G and ε. The same concerns Definition 3 and its key statement (16). But sometimes specificity of Definition 3 nonetheless makes it irresolute for a single point solution in (16). Namely that was a cause for accepting the interval solution $r^* \in (0.075;\ 0.3)$ in the exemplification with 4800-250-26-P for TM6080I classification.

CONCLUSIONS

In the problem of 26 TM6080I classification by 2LP, the CEP minimization relative gain is about a third. A best-trained classifier produces CEP not greater than 0.19 % over TM6080I, and not greater than 0.29 % over PDTM6080I. The stated framework hereby provides the CEP minimization by optimizing the 2LP classifier single PDTM6080I-training parameter.

An aggregate of other training parameters (two and more) of the classifier can be considered for optimization as well. Any rigorous framework for such multivariable optimization shall include operations from item 1 through item 4. The operations from item 8 through item 13 shall regard the corresponding multiplicity of variables. Particularly for 2LP, SHL size and integers C, F, Q_{pass} could be optimized for CEP minimization. Certainly, optimizing each of them singly is worse for CEP minimization than optimizing an aggregate of them. In defiance of statistical hugeness, multivariable optimization must help in shortening the training process duration on average. And the corresponding framework should have an effective operation item for plotting and extremizing the piecewise linear hypersurface (function of two or more variables) with values of the classifier output parameter.

ACKNOWLEDGEMENTS

The work is technically supported by the Center of Parallel Computations of Khmelnitskiy National University (Khmelnitskiy, Ukraine).

REFERENCES

1. Axinte D. A. Approach into the use of probabilistic neural networks for automated classification of tool malfunctions in broaching / D. A. Axinte // International Journal of Machine Tools and Manufacture. – 2006. – Volume 46, Issue 12–13. – P. 1445–1448. DOI: 10.1016/j.ijmachtools.2005.09.017
2. Fukushima K. Increasing robustness against background noise: Visual pattern recognition by a neocognitron / K. Fukushima // Neural Networks. – 2011. – Volume 24, Issue 7. – P. 767–778. DOI: 10.1016/j.neunet.2011.03.017
3. On the use of small training sets for neural network-based characterization of mixed pixels in remotely sensed hyperspectral images / [J. Plaza, A. Plaza, R. Perez, P. Martinez] // Pattern Recognition. – 2009. – Volume 42, Issue 11. – P. 3032–3045. DOI: 10.1016/j.patcog.2009.04.008
4. Siniscalchi S. M. Exploiting deep neural networks for detection-based speech recognition / S. M. Siniscalchi, D. Yu,

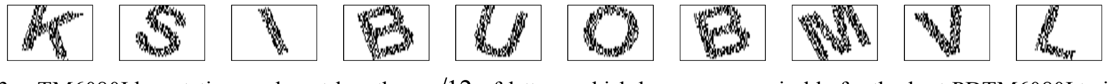

Figure 3 – TM6080I by rotation angle not less than $\pi/12$ of letters, which became recognizable for the best PDTM6080I-trained 4800-250-26-P by (24) under the set $\{C,\ F,\ Q_{\mathrm{pass}}\} = \{2,\ 8,\ 75\}$

L. Deng, C.-H. Lee // Neurocomputing. – 2013. – Volume 106. – P. 148 – 157. DOI: 10.1016/j.neucom.2012.11.008

5. Arulampalam G. A generalized feedforward neural network architecture for classification and regression / G. Arulampalam, A. Bouzerdoum // Neural Networks. – 2003. – Volume 16, Issue 5–6. – P. 561 – 568. DOI: 10.1016/S0893-6080(03)00116-3
6. Multi-column deep neural network for traffic sign classification / [D. Cireşan, U. Meier, J. Masci, J. Schmidhuber] // Neural Networks. – 2012. – Volume 32. – P. 333–338. DOI: 10.1016/j.neunet.2012.02.023
7. Fukushima K. Artificial vision by multi-layered neural networks: Neocognitron and its advances / K. Fukushima // Neural Networks. – 2013. – Volume 37. – P. 103–119. DOI: 10.1016/j.neunet.2012.09.016
8. An efficient hidden layer training method for the multilayer perceptron / [C. Yu, M. T. Manry, J. Li, P. L. Narasimha] // Neurocomputing. – 2006. – Volume 70, Issue 1–3. – P. 525–535. DOI: 10.1016/j.neucom.2005.11.008
9. Comparing evolutionary hybrid systems for design and optimization of multilayer perceptron structure along training parameters / [P. A. Castillo, J. J. Merelo, M. G. Arenas, G. Romero] // Information Sciences. – 2007. – Volume 177, Issue 14. – P. 2884–2905. DOI: 10.1016/j.ins.2007.02.021
10. Hoi K. I. Improvement of the multilayer perceptron for air quality modelling through an adaptive learning scheme / K. I. Hoi, K. V. Yuen, K. M. Mok // Computers & Geosciences. – 2013. – Volume 59. – P. 148–155. DOI: 10.1016/j.cageo.2013.06.002
11. Yuan H. C. A method for estimating the number of hidden neurons in feed-forward neural networks based on information entropy / H. C. Yuan, F. L. Xiong, X. Y. Huai // Computers and Electronics in Agriculture. – 2003. – Volume 40, Issue 1–3. – P. 57–64. DOI: 10.1016/S0168-1699(03)00011-5
12. Kathirvalavakumar T. Neighborhood based modified backpropagation algorithm using adaptive learning parameters for training feedforward neural networks / T. Kathirvalavakumar, S. Jeyaseeli Subavathi // Neurocomputing. – 2009. – Volume 72, Issue 16–18. – P. 3915–3921. DOI: 10.1016/j.neucom.2009.04.010
13. McLoone S. Improving neural network training solutions using regularisation / S. McLoone, G. Irwin // Neurocomputing. – 2001. – Volume 37, Issue 1–4. – P. 71–90. DOI: 10.1016/S0925-2312(00)00314-3
14. Kuzmanovski I. Counter-propagation neural networks in Matlab / I. Kuzmanovski, M. Novič // Chemometrics and Intelligent Laboratory Systems. – 2008. – Volume 90, Issue 1. – P. 84–91. DOI: /10.1016/j.chemolab.2007.07.003
15. Ballabio D. A MATLAB toolbox for Self Organizing Maps and supervised neural network learning strategies / D. Ballabio, M. Vasighi // Chemometrics and Intelligent Laboratory Systems. – 2012. – Volume 118. – P. 24–32. DOI: 10.1016/j.chemolab.2012.07.005
16. Romanuke V. V. Setting the hidden layer neuron number in feedforward neural network for an image recognition problem under Gaussian noise of distortion / V. V. Romanuke // Computer and Information Science. – 2013. – Volume 6, No. 2. – P. 38–54. DOI: 10.5539/cis.v6n2p38
17. On-line neural training algorithm with sliding mode control and adaptive learning rate / [A. Nied, S. I. Jr. Seleme, G. G. Parma, B. R. Menezes] // Neurocomputing. – 2007. – Volume 70, Issue 16–18. – P. 2687–2691. DOI: 10.1016/j.neucom.2006.07.019
18. Yoo S. J. Indirect adaptive control of nonlinear dynamic systems using self recurrent wavelet neural networks via adaptive learning rates / S. J. Yoo, J. B. Park, Y. H. Choi // Information Sciences. – 2007. – Volume 177, Issue 15. – P. 3074–3098. DOI: 10.1016/j.ins.2007.02.009

Article was submitted 23.05.2014.
After revision 08.10.2014.

Романюк В. В.

Д-р техн. наук, профессор кафедры прикладной математики и социальной информатики, Хмельницкий национальный университет, Украина

СТРУКТУРА ДЛЯ ОПТИМИЗАЦИИ ОТДЕЛЬНОГО ОБУЧАЮЩЕГО ПАРАМЕТРА КЛАССИФИКАТОРА НА ПРИМЕРЕ ОБУЧЕНИЯ ДВУХСЛОЙНОГО ПЕРСЕПТРОНА В ЗАДАЧЕ КЛАССИФИКАЦИИ ПОВЕРНУТЫХ ИЗОБРАЖЕНИЙ ФОРМАТА 60-НА-80

Разрабатывается структура 13-этапного плана для оптимизации отдельного обучающего параметра классификатора. Формально задача состоит в нахождении глобального экстремума (преимущественно минимума) функции, являющейся некоторым выходным параметром классификатора в зависимости от его отдельного параметра обучения. Связывая теорию плана с практикой, типом классификатора избрано двухслойный персептрон. Его входными объектами являются монохромные изображения среднего формата с несколькими тысячами независимых признаков. По представленной структуре программной средой избрано MATLAB, обладающей мощным инструментом Neural Network Toolbox. Учитывая стохастичность минимизируемой функции, определяется статистическая ε-стабильность ее оценки по конечному множеству данных. Эти данные достаются по пакетному тестированию обученного классификатора. В качестве примера использования представленной структуры плана оптимизируется соотношение среднеквадратичных отклонений пиксельных искажений и поворотов для обучения двухслойного персептрона с целью классифицировать монохромные изображения формата 60-на-80 увеличенных 26 заглавных букв английского алфавита. Целью является нахождение определенного соотношения среднеквадратичных отклонений пиксельных искажений и поворотов в процессе обучения для того, чтобы обеспечить минимум процента ошибок классификации. Относительный выигрыш такой оптимизации составляет около одной трети. Разработанная структура может быть применена также и для оптимизации классификатора с многими переменными, касательно чего отмечается, какие именно действия плана должны учитывать множественность переменных.

Ключевые слова: оптимизация параметра обучения классификатора, статистическое оценивание, план оптимизации, двухслойный персептрон, процент ошибок классификации, классификация объектов с поворотами, монохромное изображение, соотношение среднеквадратичных отклонений пиксельных искажений и поворотов, обучающее множество.

Романюк В. В.

Д-р техн. наук, професор кафедри прикладної математики та соціальної інформатики, Хмельницький національний університет, Україна

СТРУКТУРА ДЛЯ ОПТИМІЗАЦІЇ ОКРЕМОГО НАВЧАЛЬНОГО ПАРАМЕТРА КЛАСИФІКАТОРА НА ПРИКЛАДІ НАВЧАННЯ ДВОШАРОВОГО ПЕРСЕПТРОНУ В ЗАДАЧІ КЛАСИФІКАЦІЇ ПОВЕРНУТИХ ЗОБРАЖЕНЬ ФОРМАТУ 60-НА-80

Розробляється структура 13-етапного плану для оптимізації окремого навчального параметра класифікатора. Формально задача полягає у знаходженні глобального екстремуму (переважно мінімуму) функції, котра є певним вихідним параметром класифікатора залежно від його окремого параметра навчання. Пов'язуючи теорію плану з практикою, типом класифікатора обрано двошаровий персептрон. Його вхідними об'єктами є монохромні зображення середнього формату з декількома тисячами незалежних ознак. За поданою структурою програмним середовищем обрано MATLAB, котре має потужний інструмент Neural Network Toolbox. Зважаючи на стохастичність функції, що мінімізується, означається статистична ε-стабільність її оцінки за скінченною множиною даних. Ці дані дістаються за пакетним тестуванням навченого класифікатора. Для прикладу використання поданої структури плану оптимізується співвідношення середньоквадратичних відхилень піксельних спотворень і поворотів для навчання двошарового персептрону з метою класифікувати монохромні зображення формату 60-на-80 збільшених 26 великих літер англійського алфавіту. Ціллю є знаходження певного співвідношення середньоквадратичних відхилень піксельних спотворень і поворотів у процесі навчання для того, щоб забезпечити мінімум відсотка помилок класифікації. Відносний виграш такої оптимізації складає близько третини. Розроблена структура може бути застосована також і для оптимізації класифікатора з багатьма змінними, у відношенні чого зазначається, які саме дії плану мають врахувати множинність змінних.

Ключові слова: оптимізація параметра навчання класифікатора, статистичне оцінювання, план оптимізації, двошаровий персептрон, відсоток помилок класифікації, класифікація об'єктів з поворотами, монохромне зображення, співвідношення середньоквадратичних відхилень піксельних спотворень і поворотів, навчальна множина.

REFERENCES

1. Axinte D. A. Approach into the use of probabilistic neural networks for automated classification of tool malfunctions in broaching, *International Journal of Machine Tools and Manufacture,* 2006, Volume 46, Issue 12–13, pp. 1445–1448. DOI: 10.1016/j.ijmachtools.2005.09.017
2. Fukushima K. Increasing robustness against background noise: Visual pattern recognition by a neocognitron, *Neural Networks,* 2011, Volume 24, Issue 7, pp. 767–778. DOI: 10.1016/j.neunet.2011.03.017
3. Plaza J., Plaza A., Perez R., Martinez P. On the use of small training sets for neural network-based characterization of mixed pixels in remotely sensed hyperspectral images, Pattern Recognition, 2009, Volume 42, Issue 11, pp. 3032–3045. DOI: 10.1016/j.patcog.2009.04.008
4. Siniscalchi S. M., Yu D., Deng L., Lee C.-H. Exploiting deep neural networks for detection-based speech recognition, *Neurocomputing,* 2013, Volume 106, pp. 148–157. DOI: 10.1016/j.neucom.2012.11.008
5. Arulampalam G., Bouzerdoum A. A generalized feedforward neural network architecture for classification and regression, *Neural Networks,* 2003, Volume 16, Issue 5–6, pp. 561–568. DOI: 10.1016/S0893-6080(03)00116-3
6. Cireşan D., Meier U., Masci J., Schmidhuber J. Multi-column deep neural network for traffic sign classification, *Neural Networks,* 2012, Volume 32, pp. 333–338. DOI: 10.1016/j.neunet.2012.02.023
7. Fukushima K. Artificial vision by multi-layered neural networks: Neocognitron and its advances, *Neural Networks,* 2013, Volume 37, pp. 103–119. DOI: 10.1016/j.neunet.2012.09.016
8. Yu C., Manry M. T., J. Li, Narasimha P. L. An efficient hidden layer training method for the multilayer perceptron, *Neurocomputing*, 2006, Volume 70, Issue 1–3, pp. 525–535. DOI: 10.1016/j.neucom.2005.11.008
9. Castillo P. A., Merelo J. J., Arenas M. G., Romero G. Comparing evolutionary hybrid systems for design and optimization of multilayer perceptron structure along training parameters, *Information Sciences,* 2007, Volume 177, Issue 14, pp. 2884–2905. DOI: 10.1016/j.ins.2007.02.021
10. Hoi K. I., Yuen K. V., Mok K. M. Improvement of the multilayer perceptron for air quality modelling through an adaptive learning scheme, *Computers & Geosciences*, 2013, Volume 59, pp. 148–155. DOI: 10.1016/j.cageo.2013.06.002
11. Yuan H. C., Xiong F. L., Huai X. Y. A method for estimating the number of hidden neurons in feed-forward neural networks based on information entropy, *Computers and Electronics in Agriculture,* 2003, Volume 40, Issue 1–3, pp. 57–64. DOI: 10.1016/S0168-1699(03)00011-5
12. Kathirvalavakumar T., Jeyaseeli Subavathi S. Neighborhood based modified backpropagation algorithm using adaptive learning parameters for training feedforward neural networks, Neurocomputing, 2009, Volume 72, Issue 16–18, pp. 3915–3921. DOI: 10.1016/j.neucom.2009.04.010
13. McLoone S., Irwin G. Improving neural network training solutions using regularisation, *Neurocomputing,* 2001, Volume 37, Issue 1–4, pp. 71–90. DOI: 10.1016/S0925-2312(00)00314-3
14. Kuzmanovski I., Noviи M. Counter-propagation neural networks in Matlab, *Chemometrics and Intelligent Laboratory Systems,* 2008, Volume 90, Issue 1, pp. 84–91. DOI: 10.1016/j.chemolab.2007.07.003
15. Ballabio D., Vasighi M. A MATLAB toolbox for Self Organizing Maps and supervised neural network learning strategies, *Chemometrics and Intelligent Laboratory Systems,* 2012, Volume 118, pp. 24–32. DOI: 10.1016/j.chemolab.2012.07.005
16. Romanuke V. V. Setting the hidden layer neuron number in feedforward neural network for an image recognition problem under Gaussian noise of distortion, Computer and Information Science, 2013, Volume 6, No. 2, pp. 38–54. DOI: 10.5539/cis.v6n2p38
17. Nied A., Seleme S. I. Jr., Parma G. G., Menezes B. R. On-line neural training algorithm with sliding mode control and adaptive learning rate, *Neurocomputing*, 2007, Volume 70, Issue 16–18, pp. 2687–2691. DOI: 10.1016/j.neucom.2006.07.019
18. Yoo S. J., Park J. B., Choi Y. H. Indirect adaptive control of nonlinear dynamic systems using self recurrent wavelet neural networks via adaptive learning rates, Information Sciences, 2007, Volume 177, Issue 15, pp. 3074–3098. DOI: 10.1016/j.ins.2007.02.009