UDC 004.032.26::004.93

V.V. ROMANUKE
Khmelnitskiy National University

OPTIMAL PIXEL-TO-TURN-SCALE STANDARD DEVIATIONS RATIO FOR TRAINING 2-LAYER PERCEPTRON ON TURNED-SCALED OBJECTS WITH DISTRIBUTION-CONSISTENT FEATURE DISTORTION IN CLASSIFYING TURNED-SCALED OBJECTS

There is studied a problem of turned-scaled objects classification. The object model is the letter of English alphabet, which is monochrome 60-by-80-image. The classifier is 2-layer perceptron trained on turned-scaled images with normally distributed pixel distortion. The relationship among turning-scaling distortion intensities and pixel distortion intensity is regulated by pixel-to-turn-scale standard deviations ratio. For decreasing classification error percentage, the ratio is optimized. The optimal ratio is evaluated as the segment as well, where a graph of classification error percentage function has a cavity. The best-trained-under-the-optimal-ratio classifier makes errors no greater than 1.004 %.

Keywords: automatization, turned-scaled objects, object classification, neocognitron, perceptron, monochrome images, pixel distortion, turning distortion intensity, scaling distortion intensity, training set, pixel-to-turn-scale standard deviations ratio, classification error percentage.

В. В. РОМАНЮК
Хмельницький національний університет

ОПТИМАЛЬНЕ ВІДНОШЕННЯ СЕРЕДНЬОКВАДРАТИЧНИХ ВІДХИЛЕНЬ ПІКСЕЛЬНИХ СПОТВОРЕНЬ І СПОТВОРЕНЬ ПОВОРОТАМИ ТА МАСШТАБУВАННЯМ ДЛЯ НАВЧАННЯ 2-ШАРОВОГО ПЕРСЕПТРОНА НА ПОВЕРНУТИХ І МАСШТАБОВАНИХ ОБ'ЄКТАХ З УЗГОДЖЕНИМИ ЗА РОЗПОДІЛОМ СПОТВОРЕННЯМИ ОЗНАК У КЛАСИФІКАЦІЇ ПОВЕРНУТИХ І МАСШТАБОВАНИХ ОБ'ЄКТІВ

Досліджується задача класифікації повернутих і масштабованих об'єктів. Моделлю об'єкта виступає літера англійського алфавіту, котра представляє собою монохромне зображення формату 60 на 80. Класифікатором є 2-шаровий персептрон, що навчається на повернутих і масштабованих зображеннях з нормально розподіленими піксельними спотвореннями. Співвідношення між інтенсивностями спотворень поворотів і масштабування та інтенсивністю піксельних спотворень регулюється відношенням середньоквадратичних відхилень піксельних спотворень і спотворень поворотами та масштабуванням. Для зменшення відсоткового рівня помилок це відношення оптимізується. Оптимальне відношення оцінюється також як і відрізок, де графік відсоткового рівня помилок має западину. Найкращий класифікатор, навчений за оптимального відношення, робить помилки, що не перевищують 1.004 %.

Ключові слова: автоматизація, повернуті і масштабовані об'єкти, класифікація об'єктів, неокогнітрон, персептрон, монохромні зображення, піксельні спотворення, інтенсивність спотворень поворотами, інтенсивність спотворень масштабуванням, навчальна вибірка, відношення середньоквадратичних відхилень піксельних спотворень і спотворень поворотами та масштабуванням, відсотковий рівень помилок.

Problem of classification under distribution-inconsistency in object distortions

Nowadays is tightly connected with swift information flow. A huge part in this flow is automatization systems functioning, built on computer vision technique. After being detected and tracked, the object has to be classified to one of $N_{\text{class}} \in \mathbb{N} \backslash \{1\}$ classes [1]. The classification fundamental [1, 2] is in describing the object as an N-dimensional real-valued matrix $\mathbf{B} = [b_J]_{\mathscr{L}}$ of the format $\mathscr{L} = \bigtimes_{d=1}^{N} L_d$ and subscript J, having L_d features in its d-th dimension. Certainly, for flat objects $N \geqslant 2$, for solids $N \geqslant 3$, and solids in motions always are presented in four or more dimensions. If total number of object features $\prod_{d=1}^{N} L_d$ and integer N_{class} aren't great (not greater than 10, roughly), mostly classifiers are based on using the principle of minimizing the distance between the tracked object and N_{class} pattern objects (PO), or on feature-by-feature comparisons between the tracked object and N_{class} PO, or on decision trees [3, 4]. For objects with great number of features (tens, hundreds or even thousands) the neural network approximators are needed [5].

Multilayer perceptron performance is magnificent when the object at the perceptron classifier input, which naturally differs from those N_{class} PO, can be presented as one of the patterned objects, distorted in a part of its $\prod_{d=1}^{N} L_d$ features, and this feature distortion is statistically distribution-consistent. Unfortunately, these distribution-consistent feature distortions (DCFD) occur rarely. In real-time flow tracking or monitoring, the N-dimensional object by $N \in \{1, 2, 3\}$ may appear rotated (turned at a plane or space-solid angle), scaled (linearly and non-linearly), shifted towards each of N dimensions, mirrored to left or right along each of N dimensions, and so on. Such distribution-inconsistent feature distortions (DICFD) are brilliantly handled with more complicated neural networks — of hierarchical and convolutional type (cognitrons and neocognitrons). However, these complicated neural networks consume much of memory and processor resources. This undesirably delays the classification process, and in rapid flow tracking systems the tracker will only contour the objects, not keeping pace with classifying them and queuing up.

Lightening and acceleration in classifying objects with DICFD

Problem of classifying objects with DICFD that lies in consuming resources hugely and functioning lingeringly, would be solved if perceptrons as the swiftest neural networks could be effectively trained on objects with DICFD. Particularly, 2-layer perceptron (2LP) is nearly the best for classifying objects with DCFD, especially when the distortion intensity has the normal statistical distribution (all the more if there is zero expectation). Objects with DICFD, for instance, turned-scaled objects (TSO), constitute training sets, which cannot train 2LP satisfactorily [6, 7]. Weak convergence and lingering training process are symptoms of 2LP being trained on TSO [8].

Nonetheless 2LP theoretically approximates almost anything with unbounded accuracy [2, 5], if the training set is composed correctly. Being trained on TSO, 2LP clashes against multicollinearity in this distortion type. Multicollinearity generates also statistical distribution-inconsistency. And if multicollinearity in training is removed then 2LP becomes a fine classifier again. Removing the spoken multicollinearity (in general, quasi-multicollinearity or pseudo-multicollinearity) is possible through intercalating objects with DCFD into TSO. Generally speaking, intercalating objects with DCFD into objects with DICFD gives possibility to train multilayer perceptrons for classification of objects with DICFD, what must lighten and accelerate the classification process. Will investigate it on a model of TSO.

Purpose of the article and tasks for achieving it

The general 2LP performance indicator is its classification error percentage (CEP), calculated in particular as

$$p(A) = \frac{q(A)}{b \cdot N_{\text{class}}} \cdot 100 \tag{1}$$

by the number $q(A)$ of classification errors, scored at a collection of parameters-attributes A after b batches of N_{class} objects (by one representative of every class) have fed the input of 2LP. Whatever the object type is, the purpose of the classification process investigator to minimize CEP (1), solving the problem

$$A^* \in \arg\min_{A \in \boldsymbol{A}} p(A) \tag{2}$$

by the set $\boldsymbol{A}$ of all tolerable collections of the parameters-attributes within A. In the case of TSO any collection A defines rules and relationships with which DCFD are intercalated into DICFD. According to articles [6, 7, 8], which proposed a method of 2LP performance improvement in classifying TSO via training through TSO with DCFD, a collection A includes standard deviations (SD), defining the turning distortion intensity (TDI), defining the scaling distortion intensity (SDI), and defining a type of DCFD intensity eventually. There is a relationship among these SD, being an element in the collection A. With the optimized relationship by (2), 2LP performance might be improved further. In [6, 7, 8] the model of TSO was monochrome 60×80 image (M-60-80-I), noted as 60×80 matrix of zeros and ones (ZO). There were 26 M-60-80-I, corresponded to 26 enlarged English alphabet capital letters (EEACL). The background is white, and letter casts are black with white crosshatching. EEACL have a lot of generalized attributes of the real world objects (horizontal and vertical lines, squares, circles, crossings, diagonals, curves, serpentine lines, etc.) and their medium format 60×80 suits excellently for extrapolating the procedure of solving the problem (2) on other formats and object types.

So, to solve the problem (2) for turned-scaled M-60-80-I (TSM-60-80-I) with the purpose of minimizing CEP over TSM-60-80-I, there are the tasks for achieving this purpose:

1. Make a definition of general totality (GT), containing 26 PO and TSM-60-80-I, where the c-th PO is supposed to be the c-th EEACL in the list of alphabetically ordered M-60-80-I of those 26 EEACL, $c = \overline{1, 26}$.

2. Make a definition of another GT for training, including the previous one and containing TSM-60-80-I with DCFD.

3. State the configuration description of 2LP.

4. Select a method of 2LP training.

5. State the model of TSM-60-80-I with DCFD.

6. Train 2LP on TSM-60-80-I with DCFD.

7. Test the trained 2LP for evaluating the function $p(A)$ by $A \in \boldsymbol{A}$.

8. Solve the problem (2) and verify whether CEP $p(A^*)$ is minimal through re-testing the trained 2LP severer.

Certainly, during development of the article statements these generalized eight items are to be specified more exactly. The parameters-attributes collection A and the set $\boldsymbol{A}$ of such collections will be discussed circumstantially.

Definition of GT, containing 26 PO and TSM-60-80-I

An M-60-80-I has total number of features

$$\prod_{d=1}^{N} L_d = \prod_{d=1}^{2} L_d = 60 \cdot 80 = 4800 .$$

Then GT

$$G_{2^{4800}} = \left\{ \left\{ \mathbf{B}_c \right\}_{c=1}^{26}, \left\{ \tilde{\mathbf{B}}_m \right\}_{m=1}^{2^{4800}-26} \right\} \tag{3}$$

contains 26 PO $\left\{ \mathbf{B}_c = \left[b_{uv}^{\langle c \rangle} \right]_{60\times 80} \right\}_{c=1}^{26}$ and $2^{4800} - 26$ M-60-80-I $\left\{ \tilde{\mathbf{B}}_m \right\}_{m=1}^{2^{4800}-26}$ as 60×80 matrices of ZO. There are TSM-60-80-I among those $2^{4800} - 26$ M-60-80-I $\left\{ \tilde{\mathbf{B}}_m \right\}_{m=1}^{2^{4800}-26}$ within GT (3). But whether an element of the subset $\left\{ \tilde{\mathbf{B}}_m \right\}_{m=1}^{2^{4800}-26} \subset G_{2^{4800}}$ is TSM-60-80-I or not is predefined with that way, starting from the corresponding PO and continuing with its distortion event or process. Therefore, in some cases the fully black M-60-80-I may occur the super-over-enlarged (because if just over-enlarged then white crosshatching would be seen) EEACL "A" ("B", "E", "F", and others with the centered black parts), and in other cases the black M-60-80-I isn't related to EEACL. In some cases the fully white M-60-80-I may occur just the enlarged EEACL "O" ("C", "D", "L", and others having white spaces in the center of the letter M-60-80-I), and in other cases the white M-60-80-I may just be the tracked background. As soon as a PO has been turned and scaled, it changes into a TSM-60-80-I, whose matrix of ZO belongs to the subset $\left\{ \tilde{\mathbf{B}}_m \right\}_{m=1}^{2^{4800}-26} \subset G_{2^{4800}}$.

Definition of GT, including GT (3) and containing TSM-60-80-I with DCFD

While being trained, the input of 2LP is fed with samples from GT, constituted on both TSM-60-80-I and TSM-60-80-I with DCFD. The best DCFD is normal with zero expectation, whose intensity is defined with SD λ. Then another GT for training is

$$E = G_{2^{4800}} \cup \tilde{G} \tag{4}$$

by the set $\tilde{G}$ of 60×80 matrices

$$\tilde{\mathbf{G}} = \mathbf{G} + \lambda \cdot \mathbf{N} \tag{5}$$

with $\mathbf{G} \in G_{2^{4800}}$ and 60×80 matrix $\mathbf{N}$ of values of normal variate with zero expectation and unit variance (NVZEUV). GT (4) contains TSM-60-80-I with normally distributed pixel distortion (TSM-60-80-INDPD), which will be used in training.

2LP configuration description

Generally speaking, 2LP is a mapping, defined on some GT, and transferring each element of this GT into the set of N_{class} classes. This mapping has configuration

$$\mathscr{P}_0\left(\prod_{d=1}^{N} L_d ,\, N_{\text{HLN}} ,\, N_{\text{class}} ;\, f_{\text{HLTF}} ,\, f_{\text{OLTF}} \right) \tag{6}$$

by number of hidden layer neurons N_{HLN} with hidden layer transfer function f_{HLTF} and output layer transfer function f_{OLTF}. 2LP (6) for classifying M-60-80-I of $N_{\text{class}} = 26$ classes is

$$\mathscr{P}_0\left(4800,\, N_{\text{HLN}},\, 26;\, f_{\text{HLTF}},\, f_{\text{OLTF}} \right) \tag{7}$$

by integer N_{HLN} of order of hundreds or higher, depending on the object model and its distortion type. In training 2LP (6) on elements TSM-60-80-INDPD of GT (4) for classifying elements TSM-60-80-I of GT (3) there is acceptable $N_{\text{HLN}} = 300$.

2LP (7) can be configured within MATLAB environment, using accustomed MATLAB function "feedforwardnet" from Neural Network Toolbox. 2LP (7)

$$\mathscr{P}_0\left(4800,\, 300,\, 26;\, \mathscr{S},\, \mathscr{S} \right) \tag{8}$$

is initialized with the function "feedforwardnet" [8] by "logsig" transfer function (log-sigmoid transfer function or so-called "S-shaped" function) $\mathscr{S}$, having totally $4800 \cdot 300 + 300 + 26 \cdot 300 + 26 = 1448126$ weight and bias values (Figure 1).

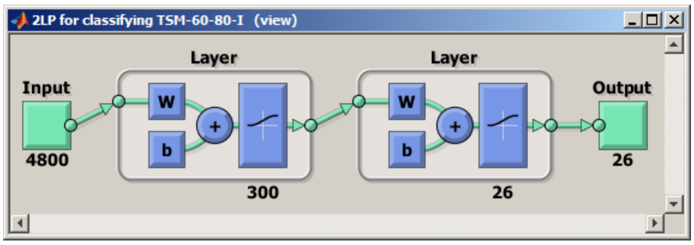

Figure 1. 2LP (8) configuration view

2LP (8) is adapted with weight and bias learning rules by Neural Network Toolbox adapt function “adaptwb”. Usefulness of 2LP (8) during training is measured with its performance function “mse” according to the sum of squared errors. Number of epochs is 15000, and let the minimum performance gradient before training is stopped be 10^{-6} as when the performance gradient becomes less than 10^{-6}, continued training is unlikely to produce significant improvements.

Selection of a MATLAB function for training of 2LP (8) on TSM-60-80-INDPD

As the transfer function $\mathcal{G}$ has derivative, then 2LP (8) can be effectively trained with backpropagation. Backpropagation training with an adaptive learning rate is implemented with MATLAB function “traingda” [8]. The function “traingda” will update 1448126 weight and bias values of 2LP (8) according to gradient descent with the adaptive learning rate [1]. Statistically, “traingda” ensures valid and effective convergence. So, this function selection is grounded on that.

Models of TSM-60-80-I and TSM-60-80-INDPD

A one TSM-60-80-INDPD is formed by (5). Mathematically a one TSM-60-80-I is formed before in two stages, although real-time transformation comes with simultaneous turning and scaling. The c-th class PO $\mathbf{B}_c$ is scaled into the c-th class EEACL $\mathbf{S}_c(z)$ of an intermediary format $V \times H$, where $z = \overline{1, Z}$ and $Z \in \mathbb{N}$ is number of portions in forming TSM-60-80-I batch. Then $\mathbf{S}_c(z)$ is turned at an angle, and the turned-scaled $V \times H$ image $\mathbf{T}_c(z)$ is re-formatted into TSM-60-80-I $\mathbf{G}_c(z)$ of the c-th class EEACL.

For z-th portion of TSM-60-80-I batch, SD

$$\eta(z) = \frac{\eta_{\max}}{Z} \cdot z \quad \text{by} \quad z = \overline{1, Z} \tag{9}$$

of SDI at $\eta_{\max} > 0$ with the value $\xi(z)$ of NVZEUV $\Xi(z)$ determines the scale coefficient

$$\varsigma(\eta(z)) = \eta(z)\xi(z) + 1 \tag{10}$$

to the scaling map

$$\sigma\left[\mathbf{B}_c, \varsigma(\eta(z))\right] = \mathbf{S}_c(z) \tag{11}$$

with the input PO $\mathbf{B}_c$. If occurs $\varsigma(\eta(z)) \leqslant 0$ then NVZEUV $\Xi(z)$ is re-raffled until $\varsigma(\eta(z)) > 0$. PO $\mathbf{B}_c$ is enlarged by $\varsigma(\eta(z))$ times via (11) if $\varsigma(\eta(z)) > 1$, and $\mathbf{B}_c$ is reduced by $\dfrac{1}{\varsigma(\eta(z))}$ times via (11) if $\varsigma(\eta(z)) < 1$. Clearly the input image $\mathbf{B}_c$ remains PO if $\varsigma(\eta(z)) = 1$. In MATLAB the map (11) is supported with MATLAB function “imresize”.

For z-th portion of TSM-60-80-I batch, SD

$$\mu(z) = \frac{\mu_{\max}}{Z} \cdot z \quad \text{by} \quad z = \overline{1, Z} \tag{12}$$

of TDI at $\mu_{\max} > 0$ with the value $\theta(z)$ of NVZEUV $\Theta(z)$ determines the angle

$$\rho(z) = \frac{180}{\pi} \cdot \mu(z)\theta(z) \tag{13}$$

in degrees, at which the scaled M-60-80-I as $\mathbf{S}_c(z)$ is turned around its center point. Matrix of ZO $\mathbf{S}_c(z)$ is processed into the turned-scaled $V \times H$ image

$$\mathbf{T}_c(z) = 1 - \tau\left[1 - \mathbf{S}_c(z), \rho(z)\right] \tag{14}$$

by the map τ, turning the input negative $1 - \mathbf{S}_c(z)$ at angle (13). $\mathbf{S}_c(z)$ through (14) is turned in counterclockwise direction if $\rho(z) > 0$, and for $\rho(z) < 0$ it is turned clockwise. Clearly for $\rho(z) = 0$ the scaled M-60-80-I as $V \times H$ matrix $\mathbf{S}_c(z)$ of ZO remains itself.

$\mathbf{T}_c(z)$ is re-formatted into TSM-60-80-I $\mathbf{G}_c(z)$ via padding or cropping the matrix. If $\varsigma(\eta(z)) > 1$ then the turned-scaled $V \times H$ image is cropped by discarding I lines and J columns in the matrix $\mathbf{T}_c(z)$, where

$$I = \left\{\left\{\overline{1, N_V}\right\}, \left\{\overline{61 + N_V, V}\right\}\right\}, \quad J = \left\{\left\{\overline{1, N_H}\right\}, \left\{\overline{81 + N_H, H}\right\}\right\}, \tag{15}$$

$$N_V = \Omega\left(\frac{V}{2}\right) - 30 + \left(\frac{1 + \operatorname{sign}\psi_V}{2} \cdot \operatorname{sign}\left|\psi_V\right|\right) \cdot \operatorname{sign}\left[\frac{V}{2} - \Omega\left(\frac{V}{2}\right)\right], \tag{16}$$

$$N_H = \Omega\left(\frac{H}{2}\right) - 40 + \left(\frac{1 + \operatorname{sign}\psi_H}{2} \cdot \operatorname{sign}\left|\psi_H\right|\right) \cdot \operatorname{sign}\left[\frac{H}{2} - \Omega\left(\frac{H}{2}\right)\right], \tag{17}$$

and $\Omega(x)$ is a function, returning the integer part of the number x, calculated by the values $\{\psi_V, \psi_H\}$ of two independent NVZEUV. These NVZEUV are raffled every time, when the function $\Omega(x)$ is applied. If $\varsigma(\eta(z))<1$ the turned-scaled $V\times H$ image $\mathbf{T}_c(z)$ is contoured rectangularly with the background white color: the matrix $\mathbf{T}_c(z)$ is padded from left and from right for

$$N_{\text{left}}=\Omega\left(\frac{80-H}{2}\right)+\left(\frac{1+\operatorname{sign}\psi_H}{2}\cdot\operatorname{sign}\left|\psi_H\right|\right)\cdot\operatorname{sign}\left[\frac{H}{2}-\Omega\left(\frac{H}{2}\right)\right] \qquad (18)$$

and

$$N_{\text{right}}=80-H-N_{\text{left}} \qquad (19)$$

columns of ones (in MATLAB the white color is coded with ones) correspondingly, and it is padded from top and from bottom for

$$N_{\text{top}}=\Omega\left(\frac{60-V}{2}\right)+\left(\frac{1+\operatorname{sign}\psi_V}{2}\cdot\operatorname{sign}\left|\psi_V\right|\right)\cdot\operatorname{sign}\left[\frac{V}{2}-\Omega\left(\frac{V}{2}\right)\right] \qquad (20)$$

and

$$N_{\text{bottom}}=60-V-N_{\text{top}} \qquad (21)$$

lines of ones correspondingly. In MATLAB the map τ in (14) is supported with MATLAB function "imrotate".

Before forming TSM-60-80-INDPD batch, matrices $\left\{\mathbf{G}_c(z)\right\}_{c=1}^{26}$ of ZO for all 26 classes TSM-60-80-I are reshaped into 4800×26 matrix $\overline{\mathbf{G}}(z)$, whose c-th column is matrix $\mathbf{G}_c(z)$, reshaped into 4800×1 matrix. And then z-th portion of TSM-60-80-INDPD batch is 4800×26 matrix

$$\tilde{\overline{\mathbf{G}}}(z)=\overline{\mathbf{G}}(z)+\lambda(z)\cdot\mathbf{N}_{26} \qquad (22)$$

by SD

$$\lambda(z)=\frac{\lambda_{\max}}{Z}\cdot z \;\text{ by }\; z=\overline{1, Z} \qquad (23)$$

of pixel distortion intensity (PDI) at $\lambda_{\max}>0$ and 4800×26 matrix $\mathbf{N}_{26}$ of values of NVZEUV. Matrix $\tilde{\overline{\mathbf{G}}}(z)$ by (22) and SD of PDI (23) includes all 26 classes TSM-60-80-INDPD, where c-th column of $\tilde{\overline{\mathbf{G}}}(z)$ corresponds to the c-th class.

Training on TSM-60-80-INDPD

For training on TSM-60-80-INDPD the training set

$$\left\{\left\{\overline{\mathbf{B}}\right\}_{d=1}^{R}, \left\{\tilde{\overline{\mathbf{G}}}(z)\right\}_{z=1}^{Z}\right\} \qquad (24)$$

feeds the input of 2LP (8) by $R+Z$ targets as $R+Z$ identity 26×26 matrices, where 26 PO $\left\{\mathbf{B}_c\right\}_{c=1}^{26}$ are reshaped into 4800×26 matrix $\overline{\mathbf{B}}$, whose c-th column is 4800×1-reshaped matrix $\mathbf{B}_c$, and $R\in□$. The training set (24) is passed through 2LP (8) for $Q\in□$ cycles. Thus 2LP (8) is trained on TSM-60-80-INDPD under parameters

$$\left\{\eta_{\max}, \mu_{\max}, \lambda_{\max}, R, Z, Q\right\}. \qquad (25)$$

Now SD of maximal SDI and TDI can be preset:

$$\eta_{\max}=0.2\,,\;\; \mu_{\max}=0.2\,. \qquad (26)$$

The relationship among these SD can be regulated with SD of maximal PDI:

$$r_{\text{PTSSD}}=\frac{\lambda_{\max}}{\mu_{\max}}=5\lambda_{\max}\,. \qquad (27)$$

Instead of the ratio (27) there could have been taken another denominator to have $\lambda_{\max}/\eta_{\max}$, but not minding equality between $\eta_{\max}$ and $\lambda_{\max}$, and so (27) is an optional parameter, involving both TDI and SDI, where relationship between them $\mu_{\max}/\eta_{\max}$ is known. Thus let (27) be called pixel-to-turn-scale standard deviations ratio (PTSSDR).

After having been trained under parameters (25), 2LP (8) transforms into 2LP

$$\mathscr{P}\left(4800, 300, 26; \mathfrak{S}, \mathfrak{S}; \eta_{\max}, \mu_{\max}, r_{\text{PTSSD}}, R, Z, Q\right). \qquad (28)$$

May the rest of parameters $\left\{R, Z, Q\right\}$ in (25) be preset to $R=2$ and $Z=8$ by sufficiently great pass integer, say, $Q=100$ to obtain fine classification capabilities of 2LP (8). Consequently, after having been trained, 2LP (28) is

$$\mathscr{P}\left(4800, 300, 26; \mathfrak{S}, \mathfrak{S}; 0.2, 0.2, r_{\text{PTSSD}}, 2, 8, 100\right). \qquad (29)$$

And this means that here the collection $A=\left\{r_{\text{PTSSD}}\right\}$ for CEP (1) and the problem (2).

Testing the trained 2LP (29) for evaluating the function $p\left(r_{\text{PTSSD}}\right)$

While 2LP (29) is tested, its input is fed with TSM-60-80-I, formed by some SD of SDI

$\eta \in [0;\, \eta_{\max}] = [0;\, 0.2]$ and some SD of TDI $\mu \in [0;\, \mu_{\max}] = [0;\, 0.2]$. As those three SD in (9), (12), (23), these SD increase simultaneously also.

Let the input of 2LP (29) be fed with $b = 200$ batches from GT (3). Let by $\eta \in [0;\, 0.2]$ and $\mu \in [0;\, 0.2]$ the number of classification errors be $q(r_{\text{PTSSD}},\, \eta,\, \mu)$. Then

$$q(r_{\text{PTSSD}}) = \frac{1}{0.2}\int_{0}^{0.2} q(r_{\text{PTSSD}},\, \eta,\, \mu)\, d\mu \approx \frac{1}{M+1}\sum_{j=0}^{M} q\left(r_{\text{PTSSD}},\, \frac{j}{5M},\, \frac{j}{5M}\right) \tag{30}$$

for $(M+1)$-pointed subset $\left\{\frac{j}{5M}\right\}_{j=0}^{M} \subset [0;\, 0.2]$ of segment $[0;\, \eta_{\max}]$ and of segment $[0;\, \mu_{\max}]$. It is sufficient to preset $M = 10$. Consequently, with (30) CEP (1) is

$$p(r_{\text{PTSSD}}) = \frac{1}{572}\cdot\sum_{j=0}^{10} q\left(r_{\text{PTSSD}},\, \frac{j}{50},\, \frac{j}{50}\right). \tag{31}$$

Let the segment of values of PTSSDR (27) be denoted as $\boldsymbol{A} = \left[r_{\text{PTSSD}}^{\langle\min\rangle};\, r_{\text{PTSSD}}^{\langle\max\rangle}\right]$. Empirically for $\lambda_{\max} > 1$ batches of TSM-60-80-INDPD are overloaded with DCFD, and for $\lambda_{\max} < 0.001$ they are felt to be underloaded. Therefore 2LP (29) is going to be trained by $r_{\text{PTSSD}} \in \left[r_{\text{PTSSD}}^{\langle\min\rangle};\, r_{\text{PTSSD}}^{\langle\max\rangle}\right] = [0.005;\, 5]$ for evaluating the function $p(r_{\text{PTSSD}})$ and solving the problem (2)

$$r_{\text{PTSSD}}^{*} \in \arg\min_{r_{\text{PTSSD}} \in [0.005;\, 5]} p(r_{\text{PTSSD}}). \tag{32}$$

Firstly let the segment of PTSSDR (27) be sampled rough, from the right side. For obtaining preliminary results faster, let take 2LP

$$\mathscr{P}(4800,\, 300,\, 26;\, \boldsymbol{\mathfrak{S}},\, \boldsymbol{\mathfrak{S}};\, 0.2,\, 0.2,\, r_{\text{PTSSD}},\, 2,\, 8,\, 25). \tag{33}$$

instead of 2LP (29). Thus there is PTSSDR segment subset

$$\{0.5 + 0.25i\}_{i=0}^{18} \subset \left[r_{\text{PTSSD}}^{\langle\min\rangle};\, r_{\text{PTSSD}}^{\langle\max\rangle}\right] = [0.005;\, 5] \tag{34}$$

and for each of 19 points in (34) the value (31) for 2LP (33) is calculated (Figure 2).

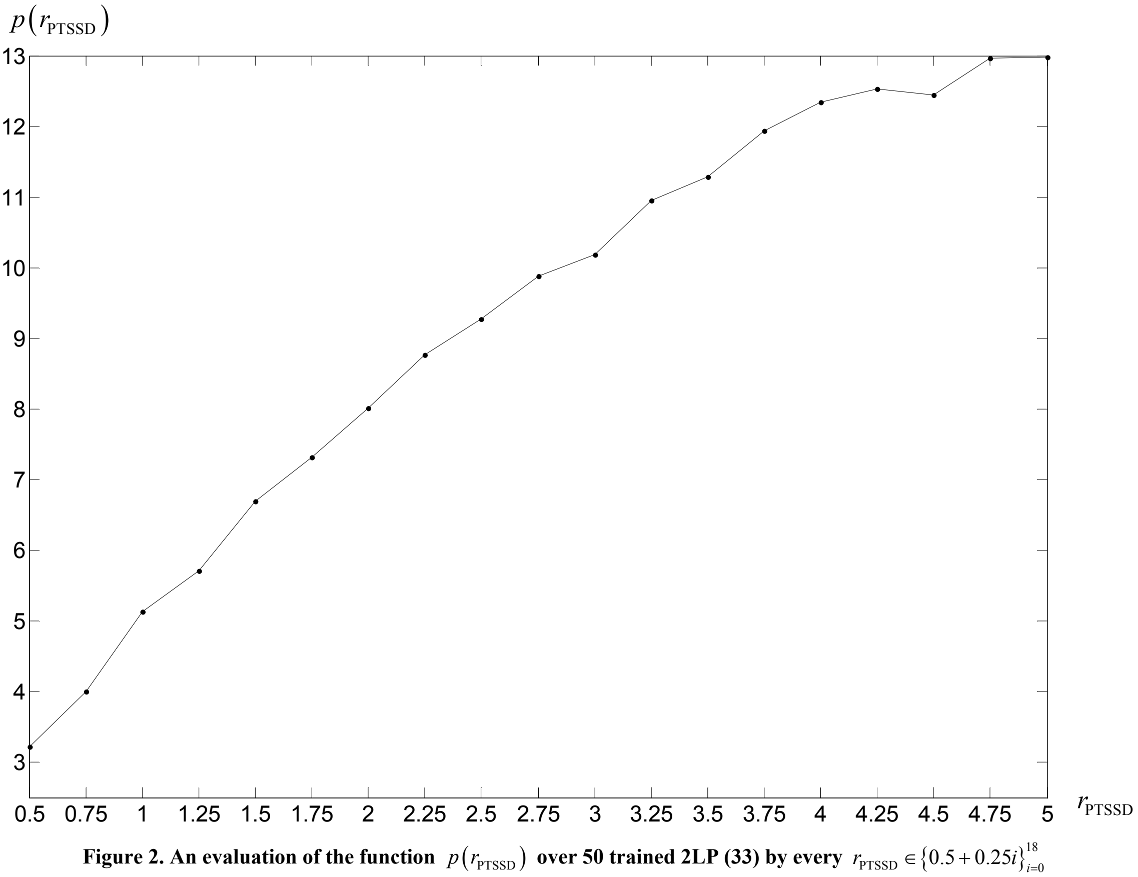

Figure 2. An evaluation of the function $p(r_{\text{PTSSD}})$ over 50 trained 2LP (33) by every $r_{\text{PTSSD}} \in \{0.5 + 0.25i\}_{i=0}^{18}$

Obviously, Figure 2 leaves vague notion about the minimum of the function $p(r_{\text{PTSSD}})$ on the segment

$\left[0.5;\,5\right]$. Nevertheless it's clear that the problem (32) is equivalent to the problem

$$r^{*}_{\text{PTSSD}} \in \arg \min_{r_{\text{PTSSD}} \in [0.005;\,0.75]} p\left(r_{\text{PTSSD}}\right), \tag{35}$$

whatever Q is. Then the function $p\left(r_{\text{PTSSD}}\right)$ for 2LP (33) is re-evaluated finer on PTSSDR subsegment $\left[0.005;\,0.75\right]$ subsets (Figure 3)

$$\left\{\left\{0.005+0.005i\right\}_{i=0}^{9},\,\left\{0.1+0.05i\right\}_{i=0}^{8},\,0.75\right\} \subset \left[0.005;\,0.75\right] \subset \left[0.005;\,5\right],$$

$$\left\{\left\{0.005+0.005i\right\}_{i=0}^{19},\,\left\{0.1+0.05i\right\}_{i=1}^{13}\right\} \subset \left[0.005;\,0.75\right] \subset \left[0.005;\,5\right]. \tag{36}$$

Those re-evaluations may be considered as zooms in the function $p\left(r_{\text{PTSSD}}\right)$ on PTSSDR subsegment $\left[0.005;\,0.75\right] \subset \left[0.005;\,5\right]$.

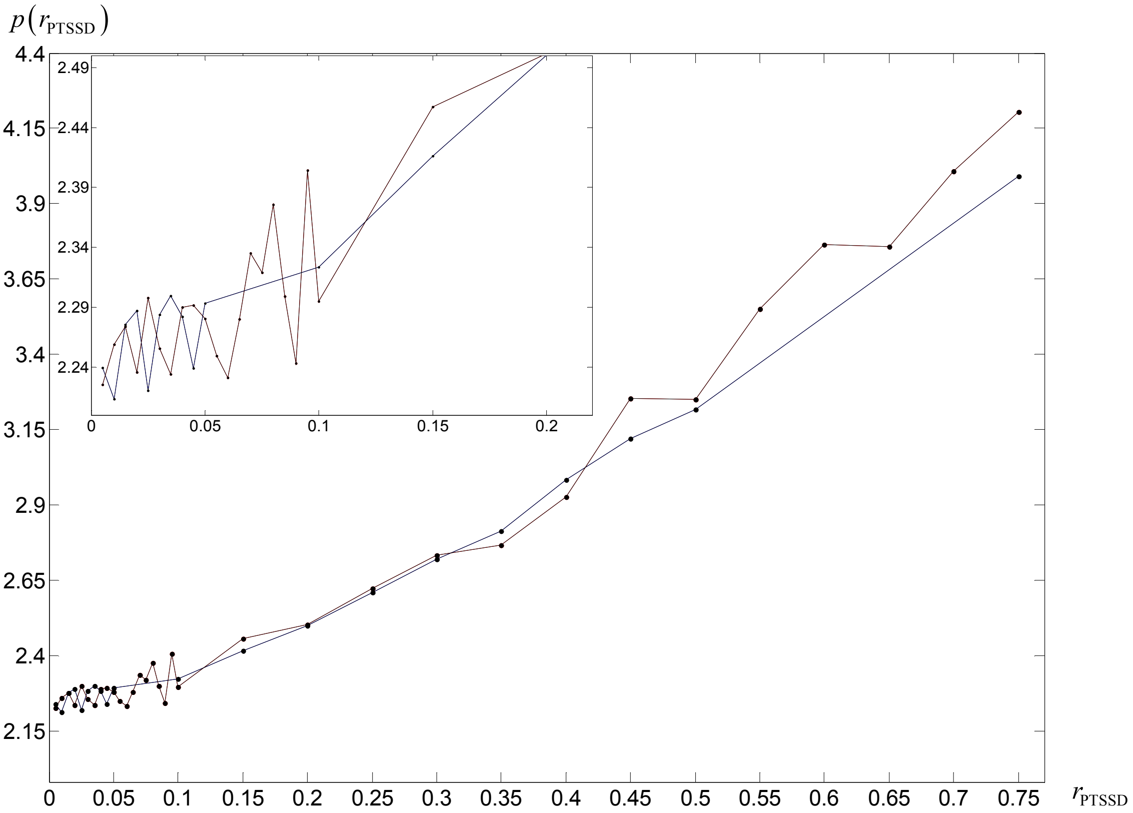

Figure 3. Re-evaluations of the function $p\left(r_{\text{PTSSD}}\right)$ for trained 2LP (33)

Unexpectedly, Figure 3 has shown that $r^{*}_{\text{PTSSD}} < 0.2$, although this is just for $Q = 25$. And 0.005-sampling is redundant. Nonetheless, re-evaluations of $p\left(r_{\text{PTSSD}}\right)$ for 2LP (33) narrow the problem (35) to the problem

$$r^{*}_{\text{PTSSD}} \in \arg \min_{r_{\text{PTSSD}} \in [0.005;\,0.4]} p\left(r_{\text{PTSSD}}\right). \tag{37}$$

For trained 2LP (29) subsequently, Figure 4 contains evaluation of the function $p\left(r_{\text{PTSSD}}\right)$ on PTSSDR subsegment $\left[0.005;\,0.4\right]$ subset

$$\left\{\left\{0.01+0.01i\right\}_{i=0}^{9},\,\left\{0.1+0.1i\right\}_{i=1}^{3}\right\} \subset \left[0.005;\,0.4\right] \subset \left[0.005;\,0.75\right] \subset \left[0.005;\,5\right] \tag{38}$$

along with the point $r_{\text{PTSSD}} = 0$. Also the upper $\hat{p}\left(r_{\text{PTSSD}}\right)$ and lower $\breve{p}\left(r_{\text{PTSSD}}\right)$ envelopes of the function $p\left(r_{\text{PTSSD}}\right)$ realizations are shown. For averaging, here 20 realizations of the function $p\left(r_{\text{PTSSD}}\right)$ on subset (38) are used. All they are shown in Figure 5.

The evaluation of the function $p\left(r_{\text{PTSSD}}\right)$ in Figure 4 shows that $r^{*}_{\text{PTSSD}} = 0.01$. The lower envelope of 20 single realizations of the function $p\left(r_{\text{PTSSD}}\right)$ in Figure 5 prompts the same. Contrariwise, $r^{*}_{\text{PTSSD}} = 0.1$ by the upper envelope. Hence, due to unstable evaluations, the problem (37) solution is $r^{*}_{\text{PTSSD}} \in \left[0.01;\,0.1\right]$.

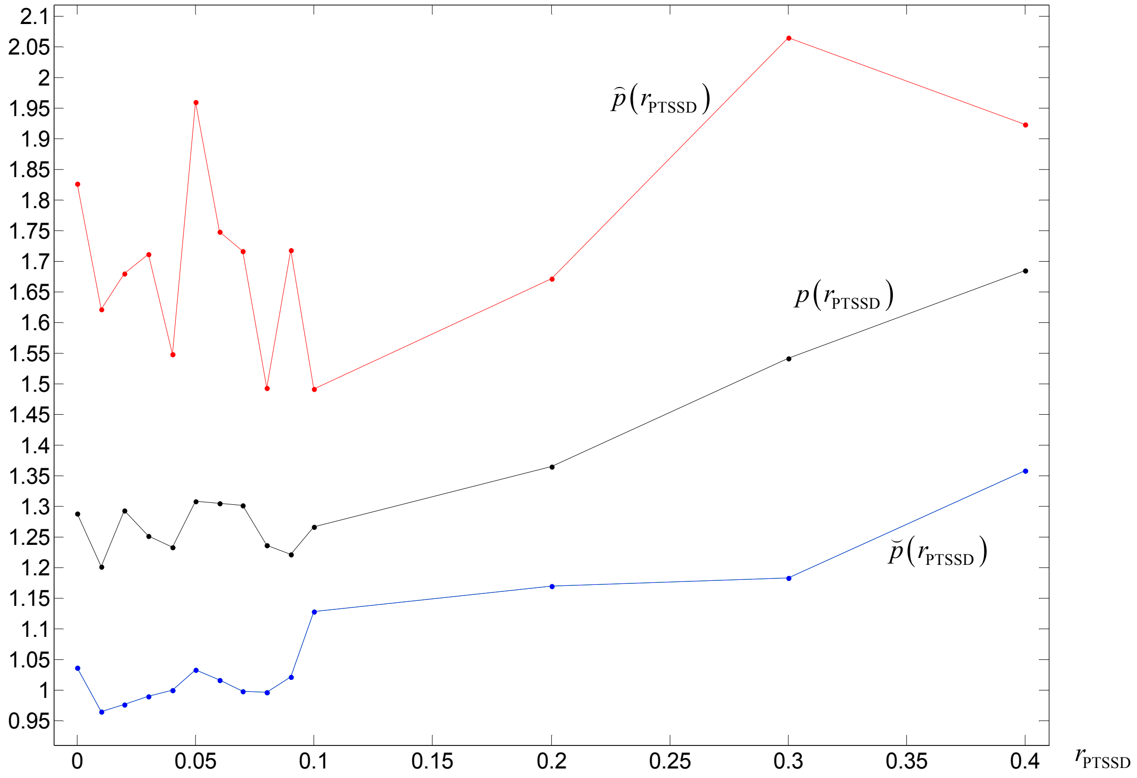

Figure 4. Evaluations of the function $p(r_{\text{PTSSD}})$ for trained 2LP (29) with the upper and lower envelopes on finite subset (38) along with the point $r_{\text{PTSSD}} = 0$

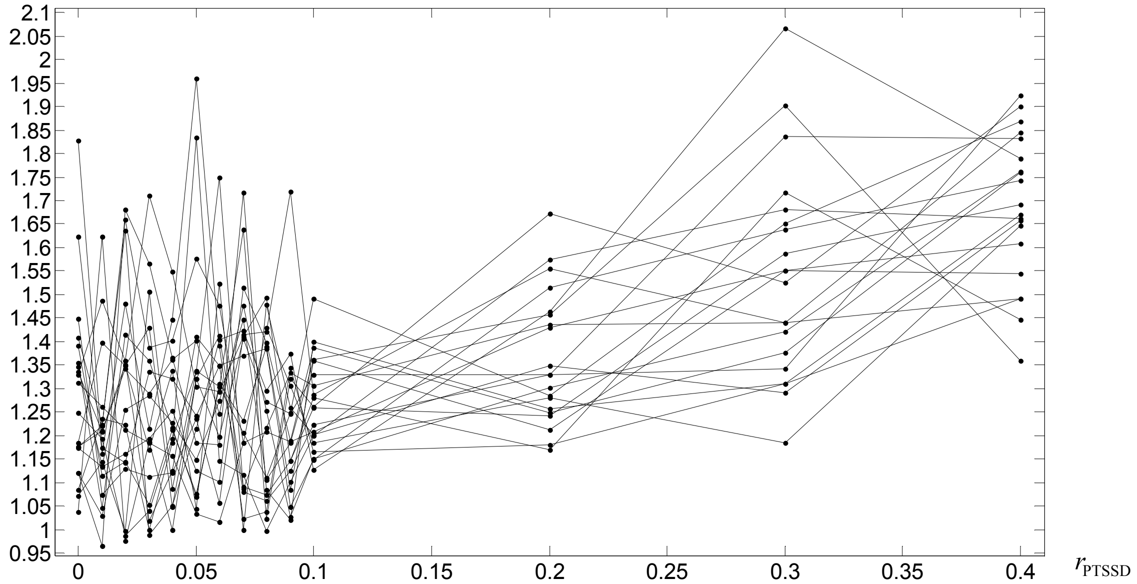

Figure 5. 20 single realizations of the function $p(r_{\text{PTSSD}})$ for trained 2LP (29) on subset (38) along with the point $r_{\text{PTSSD}} = 0$

Problem (32) solution and its verification

Figure 4 and Figure 5 both reveal a cavity within PTSSDR subsegment $[0.005;\, 0.4] \subset [0.005;\, 5]$ where $r^*_{\text{PTSSD}} \in [0.01;\, 0.1]$ in accordance with 0.01-sampling in (38). Henceforward, the point $r^*_{\text{PTSSD}} \in [0.01;\, 0.1]$ may be called 0.01-optimal PTSSDR, providing locally 0.01-minimal CEP

$$p\left(r^*_{\text{PTSSD}}\right) = p(0.01) < 0.97 \tag{39}$$

for 2LP

$$\mathscr{P}\left(4800,\, 300,\, 26;\, \mathfrak{S},\, \mathfrak{S};\, 0.2,\, 0.2,\, 0.01,\, 2,\, 8,\, 100\right) \tag{40}$$

performance. PTSSDR axis accuracy 0.01 might have been increased more, say, up to 0.005 or 0.0025, but then solving the problem (37) would have taken much more periods (realizations), what wouldn’t have been reasonable due to that the difference between (39) and

$$\min_{r_{\text{PTSSD}}\in\{[0.01;\,0.1]\}\setminus\{0.01\}} p\left(r^{*}_{\text{PTSSD}}\right) \tag{41}$$

is too small (Figure 5), and there are no any grounds to think that 0.005-optimal PTSSDR or 0.0025-optimal PTSSDR would have provided decrement of CEP, greater than that difference. However, evaluations in Figure 5 have been obtained, feeding the input of 2LP (29) with $b=200$ batches from GT (3). They may probably change, being evaluated more accurate, when b is preset greater. Let $b=2000$ to verify the point $r^{*}_{\text{PTSSD}}=0.01$ (the segment $[0.01;\,0.1]$ of PTSSDR) optimality. Figure 6 zooms in PTSSDR subsegment $[0;\,0.2]$, where the function $p\left(r_{\text{PTSSD}}\right)$ is refreshed on PTSSDR subsegment $[0;\,0.2]$ subset $\{0,\,0.01,\,0.1,\,0.2\}\subset[0;\,0.2]$. Indeed, 0.01-optimal PTSSDR remains $r^{*}_{\text{PTSSD}}\in[0.01;\,0.1]$ with CEP

$$p\left(r^{*}_{\text{PTSSD}}\right)=p(0.01)<1.004\,, \tag{42}$$

what is slightly greater than CEP (39). Figure 7 visualizes SDI and TDI by (26), which form TSM-60-80-I of EEACL, and all these TSM-60-80-I have been successfully classified with 2LP (40).

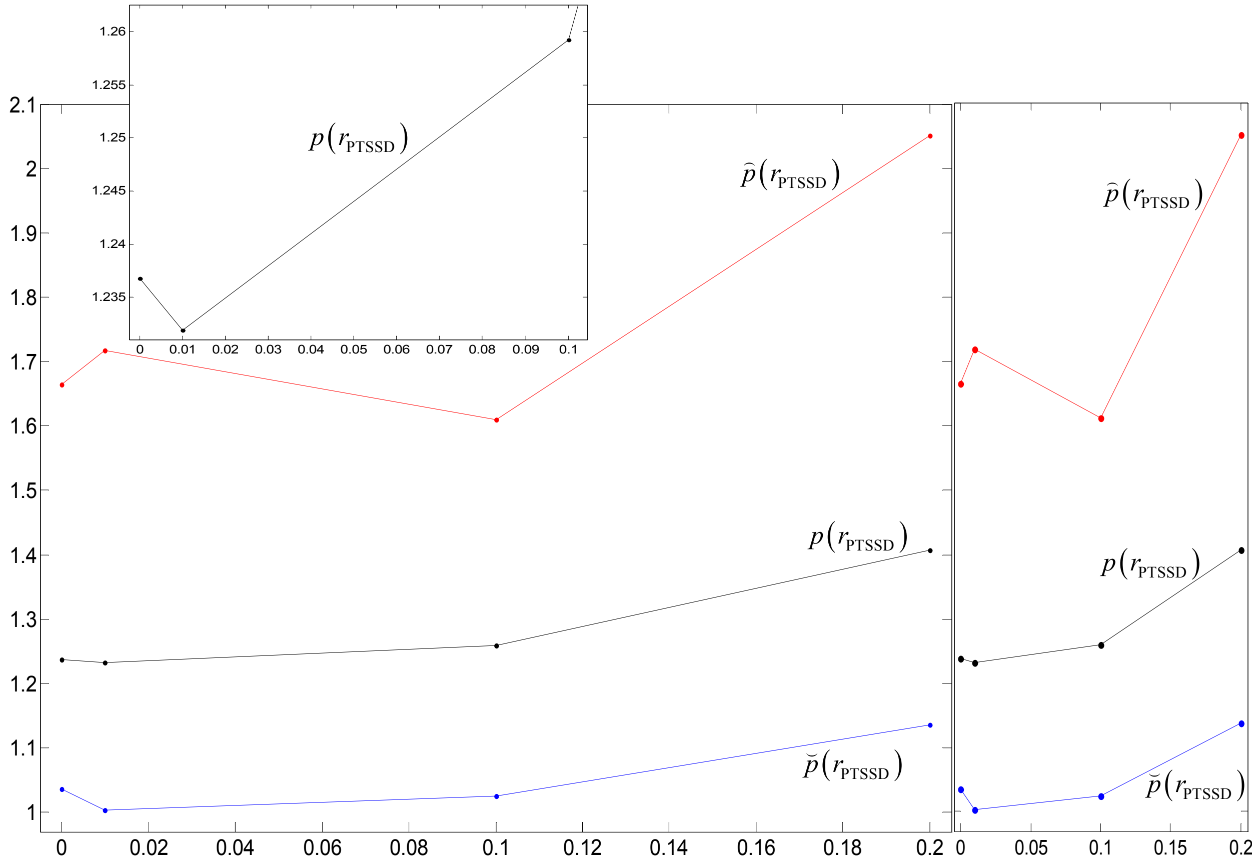

Figure 6. Re-evaluations of the function $p\left(r_{\text{PTSSD}}\right)$ cavity on 54 trained 2LP (29) by $r_{\text{PTSSD}}\in\{0,\,0.01,\,0.1,\,0.2\}$

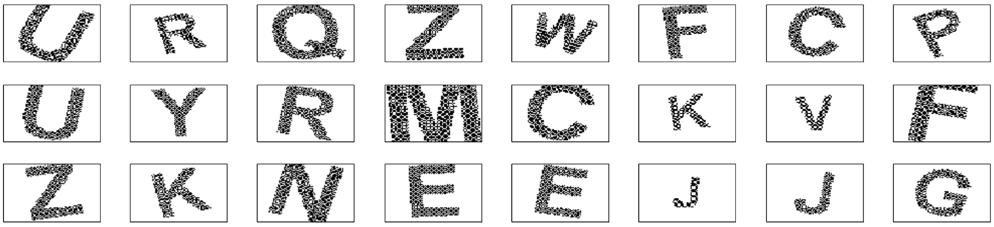

Figure 7. TSM-60-80-I of EEACL by SD of SDI and TDI (26), fed the input of 2LP (40), performing with 0.01-minimal CEP (42)

Eventually, CEP (31) for 2LP (29) has been minimized in accordance with the problem (32). Minimization has been verified with 10 times greater feed at the classifier input than the feed while the function $p\left(r_{\text{PTSSD}}\right)$ was being evaluated. Locally, PTSSDR $r^{*}_{\text{PTSSD}}=0.01$ drives 2LP (29) into 2LP (40), that effectively classifies TSO, modeled as TSM-60-80-I of 26 EEACL.

Comprehension and possibility of further 2LP performance improvement

Naturally, if by $N_{\text{class}}=26$ there were other object type of 60×80 binary format (not EEACL), noted with matrix of ZO, then the result of (32) would be nearly the same, close to $r^{*}_{\mathrm{PTSSD}} = 0.01$ or $r^{*}_{\mathrm{PTSSD}} \in [0.01;\, 0.1]$. Moreover, inasmuch as N-dimensional objects, feeding the input of 2LP, with its N-dimensional matrix $\mathbf{B} = [b_J]_{\mathscr{L}}$ of ZO is always reshaped into $\left(\prod_{d=1}^{N} L_d\right) \times 1$ matrix, then nearly the same minimum point $r^{*}_{\mathrm{PTSSD}} = 0.01$ must be for any object, having 4800 binary features. That is there can be TSO of formats 80×60, $6 \times 8 \times 100$, $12 \times 4 \times 10 \times 10$, $2 \times 2 \times 4 \times 3 \times 5 \times 10 \times 2$, etc. 2LP $\mathscr{P}\left(\prod_{d=1}^{N} L_d,\, 300,\, N_{\mathrm{class}};\, \mathfrak{S},\, \mathfrak{S};\, 0.2,\, 0.2,\, 0.01,\, 2,\, 8,\, 100\right)$ for other medium formats, where objects have a few thousands of features $\prod_{d=1}^{N} L_d$ and number N_{class} is about 20 — 35, must perform near-optimally as well.

Further improvement of 2LP performance over TSO is perceived in optimizing the integer N_{HLN}. The optimally adjusted N_{HLN} will allow to accelerate the training process, making the configuration of 2LP lighter. Of course, that noticeable minimization of CEP over TSO should be done for N_{HLN} and r_{PTSSD} simultaneously — power-computational evaluations of CEP surface $p(N_{\mathrm{HLN}},\, r_{\mathrm{PTSSD}})$ are inescapable.

References

1. Haykin S. Neural Networks: A Comprehensive Foundation, 2nd ed., Moscow, Publishing house "Williams", 2006, 1104 p.
2. Gorban A. N. Generalized approximation theorem and computational capabilities of neural networks, Siberian journal of computational mathematics, 1998, Vol. 1, N. 1, pp. 11 – 24.
3. Ghimire S., Wang H. Classification of image pixels based on minimum distance and hypothesis testing, Computational Statistics & Data Analysis, 2012, Vol. 56, Iss. 7, pp. 2273 – 2287.
4. Farid D. M., Zhang L., Rahman C. M., Hossain M. A., Strachan R. Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks, Expert Systems with Applications, 2014, Vol. 41, Iss. 4, Part 2, pp. 1937 – 1946.
5. 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.
6. Romanuke V. V. A 2-layer perceptron performance improvement in classifying 26 turned monochrome 60-by-80-images via training with pixel-distorted turned images, Research bulletin of the National Technical University of Ukraine "KPI", 2014, N. 5, pp. 55 – 62.
7. Romanuke V. V. Classification error percentage decrement of two-layer perceptron for classifying scaled objects on the pattern of monochrome 60-by-80-images of 26 alphabet letters by training with pixel-distorted scaled images, Scientific bulletin of Chernivtsi National University of Yuriy Fedkovych. Series: Computer systems and components, 2013, Vol. 4, Iss. 3, pp. 53 – 64.
8. Romanuke V. V. A 2-layer perceptron performance improvement in classifying turned-scaled monochrome 60×80 images via training through turned-scaled images with pixel distortion, Scientific bulletin of Chernivtsi National University of Yuriy Fedkovych. Series: Computer systems and components, 2013, Vol. 4, Iss. 4, pp. 56 – 65.

Література

1. Хайкин С. Нейронные сети: полный курс / Хайкин С. – [2-е изд.]. – М. : Издательский дом "Вильямс", 2006. – 1104 с.
2. Горбань А. Н. Обобщённая аппроксимационная теорема и вычислительные возможности нейронных сетей / А. Н. Горбань // Сибирский журнал вычислительной математики, 1998. – Т. 1, № 1. – С. 11 – 24.
3. Ghimire S. Classification of image pixels based on minimum distance and hypothesis testing / S. Ghimire, H. Wang // Computational Statistics & Data Analysis. – 2012. – Volume 56, Issue 7. – P. 2273 – 2287.
4. Farid D. M. Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks / D. M. Farid, L. Zhang, C. M. Rahman, M. A. Hossain, R. Strachan // Expert Systems with Applications. – 2014. – Volume 41, Issue 4, Part 2. – P. 1937 – 1946.
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.
6. Romanuke V. V. A 2-layer perceptron performance improvement in classifying 26 turned monochrome 60-by-80-images via training with pixel-distorted turned images / V. V. Romanuke // Наукові вісті НТУУ "КПІ". – 2014. – № 5. – С. 55 – 62.
7. Romanuke V. V. Classification error percentage decrement of two-layer perceptron for classifying scaled objects on the pattern of monochrome 60-by-80-images of 26 alphabet letters by training with pixel-distorted scaled images / V. V. Romanuke // Науковий вісник Чернівецького університету імені Юрія Федьковича. Серія: Комп'ютерні системи та компоненти. – Чернівці : ЧНУ, 2013. – Т. 4, вип. 3. – С. 53 – 64.
8. Romanuke V. V. A 2-layer perceptron performance improvement in classifying turned-scaled monochrome 60×80 images via training through turned-scaled images with pixel distortion / V. V. Romanuke // Науковий вісник Чернівецького університету імені Юрія Федьковича. Серія: Комп'ютерні системи та компоненти. – Чернівці : ЧНУ, 2013. – Т. 4, вип. 4. – С. 56 – 65.

Рецензія/Peer review : 13.1.2015 р. Надрукована/Printed :25.1.2015 р.
Рецензент: д.т.н., проф. Троцишин І.В.