

Optimal Training Parameters and Hidden Layer Neuron Number of Two-Layer Perceptron for Generalised Scaled Object Classification Problem

Vadim Romanuke
Khmelnitsky National University

Abstract – The research is focused on optimising two-layer perceptron for generalised scaled object classification problem. The optimisation criterion is minimisation of inaccuracy. The inaccuracy depends on training parameters and hidden layer neuron number. After its statistics is accumulated, minimisation is executed by a numerical search. Perceptron is optimised additionally by extra training. As it is done, the classification error percentage does not exceed 3 % in case of the worst scale distortion.

Keywords – Extra pass training, optimisation, scaling-proof classifier, two-layer perceptron.

I. INTRODUCTION

A problem of classifying scaled object issues partially appears due to the impossibility to avoid distortions. On the other hand, objects may have different dimensions. Thus, they are treated as scaled ones regarding the average dimensions. Generally, this is a problem of image recognition [1], [2]. Objects appear closer and farther in front of the cam. And it is difficult to adjust focal length every time. However, three-dimensional objects also occur scaled [3]. Scaling of multidimensional objects can be comprehended as scaled coloured images with metadata [4].

Classifiers must perform reliably through streams of scaled objects. Desired classifiers' characteristics are speed of classification, quick resetting, saving of disk space and memory. It is important to provide these characteristics along with increasing classification accuracy.

II. RELATED WORKS

Contemporary scaling-proof classifiers tend to be constructed on the basis of deep learning [5], [6]. Decision trees are effectively applied when the number of object features is not great [7], [8]. Except random forests [9], boosting is applicable when the number of object features is in the order of hundreds or thousands [10], [11].

In real practice, it is impossible to ensure 100 % object recognition. Even at moderate scale distortion, classification error percentage (CEP) is scarcely ever zero [1], [2], [4]. At maximal scale distortion, CEP rises up to a few percent [12]. Exact evaluation is troublesome because different tests use various benchmarks, and maximum distortion intensity differs badly. Besides, accuracy is evaluated diversely. Sometimes, it is CEP averaged over the whole range of the scale distortion. Another time, accuracy is put as CEP at the maximum scale

distortion. This maximum is uncertain, though.

Scaling is not the worst object distortion for classifier. Shifting and rotating are worse. Therefore, the scaling-proof classifier should not be too complicated and resource-intensive like deep neural networks. According to the universal approximation theorem for feedforward neural networks [13], [14], two-layer perceptron (2LP) could be applied to classify scaled objects. Advantages of 2LP are simplicity and high speed of classification [12], [15], [16]. Quick resetting and resource-saving by low CEP are believed to be reachable if 2LP is optimised by its hidden layer neuron number (HLNN) along with the training parameters [12], [17], [18]. The criterion of optimisation is minimisation of CEP, which depends on the training parameters. HLNN is also to be minimised, but the two-criterion minimisation problem may turn unsolvable. Thus, HLNN will be included into a list of variables influencing CEP. To state and solve the minimisation problem, basic formalisations and assignments follow.

III. ARTICLE GOAL AND OBJECTIVES

Proceeding from an assumption that the globally optimised 2LP can be the scaling-proof classifier, the goal is to state the process of optimisation and to obtain the optimal training parameters and HLNN of 2LP. To achieve the goal, the 12 associated objectives are to be accomplished:

1. To list a series of the training parameters.
2. To substantiate a criterion of optimisation.
3. To formalise the optimisation problem, i.e., to state it in mathematical terms.
4. To define a type of objects for benchmark classification.
5. To ascertain how bad it is scaled when it is spoken about maximum scale distortion.
6. To state a model of scale distortion.
7. To recommend how to determine ranges of variables in the objective function whose values are CEP.
8. To formalize the 2LP classifier.
9. To describe how training sets and testing sets are formed.
10. To suggest a way of accumulating statistics for CEP which depends on the listed variables and HLNN.
11. To ascertain how to solve the formalised optimisation problem by the accumulated statistics.
12. To make suggestions of improving and applying the stated optimisation technique.

The article results will be discussed in the view of dissemination for other distortion types. In conclusion, an outlook for developing

the 2LP optimisation technique will be given.

IV. TRAINING PARAMETERS

Training sets are composed of batches of pure object representatives (PORs) and distorted (scaled) samples (DSSs). The integer R is a number of replicas of POR in the training set. The integer B is a number of batches of DSS, where each batch is the POR batch distorted at definite scale distortion intensity (SDI).

By an example of PORs which are monochrome images, training sets are corrected under a pixel-to-scale standard deviation ratio (PSSDR) introduced in [12]. The correction is in importing a share of pixel distortion into DSS. This makes training sets milder for 2LP whose performance under normally distributed feature distortions (NDFD) is excellent [18], [19]. PSSDR r defines the share of NDFD.

The last training parameter Q is a number of the training set passes through 2LP. This integer might be determined by R and B . But bonds between these three integers are implicit [17], [20], [21].

V. CRITERION OF OPTIMISATION

Denote CEP by $p(N, R, B, r, Q)$ which is a function of five variables, where N is HLNN. In fact, this function is determined at some SDI that is going to be mentioned specifically. Note that only variable r is continuous and nonnegative [12], [16], [20]. The remaining ones are positive integers. Hence, the function $p(N, R, B, r, Q)$ is to be minimised on a five-dimensional lined lattice (5DLL).

VI. FORMALISATION OF THE OPTIMISATION PROBLEM

Formally, the optimisation problem is stated as

$$\mathbf{A}^* \in \arg \min_{\mathbf{A} \in \mathbf{A}} p(N, R, B, r, Q) \quad (1)$$

by the 5DLL

$$\mathbf{A} = \{[N \ R \ B \ r \ Q] \in \mathbb{R}^5 : [N \ R \ B \ Q] \in \mathbb{N}^4, r \geq 0\}.$$

The minimum point

$$\mathbf{A}^* = [N^* \ R^* \ B^* \ r^* \ Q^*]$$

may turn out to be nonsingle, and this is reflected in (1). As soon as ranges of variables are determined, the 5DLL \mathbf{A} will be re-defined bounded above.

VII. OBJECTS FOR BENCHMARK CLASSIFICATION

Taken monochrome images as POR, we have to define their size and number of classes C . The object size is the flat image format $H \times L$. For benchmark classification, these parameters should be varied in order to ensure validity of the 2LP optimisation technique.

A quite good monochrome image model (MIM) is the English alphabet capital letters (EACL) [12], [15]. Therefore, $C \leq 26$. The image height H and length L experienced in

papers [11], [12], [15], [22], [23] are taken from a medium format: $H \leq 60$ and $L \leq 80$. Thus, the start-off pattern object size can be the half of the maximum, i.e., this is 30×40 . And the minimum format to be considered is 15×20 . Formats between 15×20 and 60×80 must be taken proportionally.

An EACL, being MIM of POR, is scaled to maximum bad enlargement if the letter body exceeds $H \times L$ contour. This is about 50% enlargement. Thus, it is maximum bad reduction. However, the seeming symmetry is delusive inasmuch as 50% enlargement means enlarging by 1.5 times, but 50% reduction means reducing twice.

VIII. MODEL OF SCALE DISTORTION

In mathematical denotation, a POR is an $H \times L$ matrix. For MIM, elements of this matrix have values 0 and 1 [15], [23]. Let 1 be the value of the background colour (usually, white). For POR of the c -th class, denote its matrix by \mathbf{M}_c , $c = \overline{1, C}$. Henceforward, the c -th class POR is flagged by \mathbf{M}_c . The training set is formed in B stages. At the k -th stage, the scale coefficient (SC)

$$\zeta(\sigma_k) = \sigma_k \xi_k + 1 \quad \text{by } k = \overline{1, B} \quad (2)$$

determines SDI [12], where ξ_k is a value of standard normal distribution (SND) drawn at the k -th stage for each class separately by the scale standard deviation (SSD)

$$\sigma_k = kB^{-1} \sigma_B \quad \text{for } \sigma_B > 0. \quad (3)$$

The value ξ_k is re-drawn until $\zeta(\sigma_k) > 0$. Theoretically, if $\zeta(\sigma_k) > 1$ then the image \mathbf{M}_c is enlarged by $\zeta(\sigma_k)$ times; if $\zeta(\sigma_k) < 1$ then it is reduced by $(\zeta(\sigma_k))^{-1}$ times. By $\zeta(\sigma_k) = 1$ the image \mathbf{M}_c remains non-scaled. Practically, if SC (2) is too close to 1, SC (2) is rounded to 1 and there is no scaling effect.

Whatever SC (2) is, the scaled object is flagged by the $H \times L$ matrix $\tilde{\mathbf{M}}_c(k)$. This matrix is the result of the scaling map Θ [1], [2], [4], [12], [23]:

$$\tilde{\mathbf{M}}_c(k) = \Theta(\mathbf{M}_c, \zeta(\sigma_k)). \quad (4)$$

The scaling map Θ in (4) is defined for any $H \times L$ matrix of ones and zeros by $\zeta(\sigma_k) > 0$.

Before the matrix $\tilde{\mathbf{M}}_c(k)$ is returned, a matrix $\mathbf{S}_c(k)$ of the intermediate format $H_* \times L_*$ is obtained within the map (4). The matrix $\mathbf{S}_c(k)$ represents the scaled image \mathbf{M}_c before reverting to the initial format $H \times L$. If $\zeta(\sigma_k) > 1$ then the image $\mathbf{S}_c(k)$ is cropped: lines of their numbers

$$\left\{ \overline{1, h_*}, \overline{H+1+h_*, H_*} \right\} \quad (5)$$

and columns of their numbers

$$\left\{ \left\{ \overline{1}, \overline{L_*} \right\}, \left\{ \overline{L+1+L_*}, \overline{L_*} \right\} \right\} \quad (6)$$

in the matrix $\mathbf{S}_c(k)$ are discarded [12]. Integers

$$h_* = \eta \left[0.5(H_* - H) \right] + \left(\left[0.5 + 0.5 \operatorname{sign} \zeta_1 \right] \cdot \operatorname{sign} |\zeta_1| \right) \cdot \operatorname{sign} \left[0.5H_* - \eta(0.5H_*) \right] \quad (7)$$

and

$$l_* = \eta \left[0.5(L_* - L) \right] + \left(\left[0.5 + 0.5 \operatorname{sign} \zeta_2 \right] \cdot \operatorname{sign} |\zeta_2| \right) \cdot \operatorname{sign} \left[0.5L_* - \eta(0.5L_*) \right] \quad (8)$$

for (5) and (6) are calculated by drawing independently a couple of two values $\{\zeta_1, \zeta_2\}$ from SND, where function $\eta(x)$ returns the integer part of the number x [12], [23].

If $\zeta(\sigma_k) < 1$ then the reduced image is contoured rectangularly with the background colour: the matrix $\mathbf{S}_c(k)$ is padded from the left for

$$l_1 = \eta \left[0.5(L - L_*) \right] + \left(\left[0.5 + 0.5 \operatorname{sign} \zeta_2 \right] \cdot \operatorname{sign} |\zeta_2| \right) \cdot \operatorname{sign} \left[0.5L - \eta(0.5L_*) \right] \quad (9)$$

columns of ones and from the right for

$$l_2 = L - L_* - l_1 \quad (10)$$

columns of ones, and it is padded from the top for

$$h_1 = \eta \left[0.5(H - H_*) \right] + \left(\left[0.5 + 0.5 \operatorname{sign} \zeta_1 \right] \cdot \operatorname{sign} |\zeta_1| \right) \cdot \operatorname{sign} \left[0.5H_* - \eta(0.5H_*) \right] \quad (11)$$

lines of ones and from the bottom for

$$h_2 = H - H_* - h_1 \quad (12)$$

lines of ones [12], [23].

IX. RANGES OF VARIABLES

Rational HLNN is determined by number of object features $F = H \cdot L$, SDI maximum expressed by SSD σ_B , number of classes C , and type of objects. Based on experience in [12], [15], [22], [23], $N \in \{50, 250\}$ for $F \in \{1200, 4800\}$. Number of replicas of POR in the training set cannot be too great. Thus, $R = 1$ or $R = 2$. Clearly, $B > R$ and, thus, $B \in \{R+1, 20\}$. PSSDR $r \in [0; 0.2]$ is ample. Again, based on the said experience, $Q \in \{10, 50\}$ is enough to train 2LP satisfactorily.

By the ascertained ranges of variables, problem (1) is restated clearer:

$$\mathbf{A}^* = \left[N^* \quad R^* \quad B^* \quad r^* \quad Q^* \right] \in$$

$$\in \arg \min_{\substack{N \in \{50, 250\}, R \in \{1, 2\} \\ B \in \{R+1, 20\}, r \in [0; 0.2], Q \in \{10, 50\}}} p(N, R, B, r, Q). \quad (13)$$

Solving problem (13) requires statistics to evaluate CEP. Before realising it, the 2LP classifier is formalised, and formations of training sets and testing sets are described.

X. FORMALISATION OF THE 2LP CLASSIFIER

The 2LP algorithm lies in calculating the c_0 -th output neuron value [16], [20]

$$v_{c_0} = \lambda \left(\sum_{j=1}^N \lambda \left(\sum_{i=1}^F x_i u_{ij}^{(1)} + b_j^{(1)} \right) u_{jc_0}^{(2)} + b_{c_0}^{(2)} \right) \quad \text{at } c_0 = \overline{1, C} \quad (14)$$

by two real-valued matrices and vectors

$$\mathcal{P}(N) = \left\{ \left[u_{ij}^{(1)} \right]_{F \times N}, \left[u_{jc_0}^{(2)} \right]_{N \times C}, \left[b_j^{(1)} \right]_{1 \times N}, \left[b_{c_0}^{(2)} \right]_{1 \times C} \right\} \quad (15)$$

and the object features $\{x_i\}_{i=1}^F$, where the logarithmic sigmoid transfer function $\lambda(x) = (1 + e^{-x})^{-1}$ is used in N neurons of the hidden layer and in C neurons of the output layer [24], [25]. After values $\{v_{c_0}\}_{c_0=1}^C$ in (14) are calculated [19], 2LP classifier returns the class number c^* which is

$$c^* \in \arg \max_{c_0=1, C} v_{c_0}. \quad (16)$$

To get the class number (16) matching to the class of the object at the 2LP input, values (15) are adjusted while 2LP is trained [26], [27]. By the way, testing sets differ from training sets because it is sufficient to evaluate CEP only at the SDI maximum [15], [23], [28], [29].

XI. TRAINING SETS AND TESTING SETS

A replica of all POR is $F \times C$ matrix \mathbf{G} whose c -th column is the c -th class representative \mathbf{M}_c reshaped into F -length-column. The k -th batch of DSS is $F \times C$ matrix $\tilde{\mathbf{G}}(k)$ whose c -th column is the c -th class representative $\tilde{\mathbf{M}}_c(k)$ reshaped into F -length-column. DSSs are completed by adding NDFD:

$$\mathbf{D}(k) = \tilde{\mathbf{G}}(k) + \gamma_k \cdot \Psi_k \quad (17)$$

by NDFD standard deviation

$$\gamma_k = kB^{-1}\gamma_B \quad \text{for } \gamma_B > 0 \quad (18)$$

and $F \times C$ matrix Ψ_k of values of SND drawn for the k -th batch. A training set for a pass through 2LP is

$$\left\{ \left\{ \mathbf{G} \right\}_{r_0=1}^R, \left\{ \mathbf{D}(k) \right\}_{k=1}^B \right\} \quad (19)$$

by PSSDR $r = \gamma_B \sigma_B^{-1}$ [12]. A testing set is the matrix $\tilde{\mathbf{G}}(B)$ corresponding to the SDI maximum. CEP $p(N, R, B, r, Q)$ is evaluated by feeding the 2LP input with 400 testing sets.

XII. STATISTICS FOR CEP

The 5DLL A must be sampled across the dimension of variable r . Each point of the sampled 5DLL A is a quintuple of 2LP parameters at which the 2LP should be trained and tested no less than five times. If the sampling step is 0.01 then the number of all 2LP to be trained and tested is over 32 million (32,016,285). Be aware of the whole evaluation process that should be exercised on an assemblage of multiprocessors, ready to be parallelised. Time required for training and testing a 2LP is up to 10 minutes. Consequently, we need to run a few thousand threads. For instance, running 500 eight-core processors takes almost two months to get the function $p(N, R, B, r, Q)$ evaluated. This, nonetheless, is realisable owing to cloud services [30], [31] or clusters [32].

For $r \in \{0.01s\}_{s=0}^{20}$ and five trainings of each 2LP, the whole statistics can be stored in two five-dimensional matrices: in matrix

$$\mathbf{H}_1 = \left(z_{i_1 i_2 i_3 i_4 i_5}^{(1)} \right)_{201 \times 19 \times 21 \times 41 \times 5} \quad (20)$$

for $R=1$ and in matrix

$$\mathbf{H}_2 = \left(z_{i_1 i_2 i_3 i_4 i_5}^{(2)} \right)_{201 \times 18 \times 21 \times 41 \times 5} \quad (21)$$

for $R=2$. The dimension for R is omitted, and the fifth dimension is the training number. If data type of matrices (20) and (21) is double precision array, then 16440795 elements of (20) use 131526360 bytes, and 15575490 elements of (21) use 124603920 bytes. Nevertheless, single precision is sufficient for accumulating CEP statistics. Then, matrices (20) and (21) use 65763180 and 62301960 bytes, respectively. Grand total is 32016285 elements using less than 123 megabytes.

XIII. SOLUTION

After having accumulated the CEP statistics in matrices (20) and (21), the function $p(N, R, B, r, Q)$ is evaluated by averaging over the fifth dimension of those matrices. Minimisation is executed by a proof numerical search approach [12], [33], [34]. For classifying 30×40 EACL by $C=26$, the problem (13) solution is roughly

$$\mathbf{A}^* = [109 \quad 2 \quad 18 \quad 0.02 \quad 47], \quad (22)$$

i.e. $N^* = 109$, $R^* = 2$, $B^* = 18$, $r^* = 0.02$, $Q^* = 47$, where

$$p(109, 2, 18, 0.02, 47) < 3.9, \quad (23)$$

meaning that just one-twenty-fifth EACL is classified wrong under the worst scaling distortion.

Solution (22) is nearly actual for 48×64 EACL by $C=26$, although it is much rougher. CEP is decreased, amazingly, by 8% to the value in (23). The quasi-optimal HLNN for classifying 15×20 EACL by $C=26$ decreases to 100, while

the rest four training parameters keep.

When number of classes decreases, optimality of $Q^* = 47$ and both $R^* = 2$ and $B^* = 18$ keeps. But HLNN N^* decreases: $N^* = 88$ for classifying 15×20 EACL by $C=10$. Nonetheless, decrement of r^* is, rather, minimal (obviously, $r^* = 0.01$ or it may be still $r^* = 0.02$). And CEP is naturally decreased to 3% (the lowest CEP happened to be 2.23% at $r^* = 0.01$).

Furthermore, CEP could be decreased lower by training the 2LP with extra passes after Q^* passes. Let the maximum of passes be Q_{\max} , where $Q_{\max} > Q^*$. And let the maximum of trials to improve CEP be t_{\max} , where $t_{\max} \geq 1$. Denote by \tilde{p} the temporary CEP, being the CEP after Q^* passes. The temporary 2LP is $\tilde{\mathcal{P}}(N^*)$, which is the 2LP $\mathcal{P}(N^*)$ after Q^* passes. The best decreased CEP is p_* , being

$$p_* = \tilde{p} = p(N^*, R^*, B^*, r^*, Q^*)$$

before the first extra pass. The best 2LP is denoted by $\mathcal{P}_*(N^*)$.

The first extra pass is $q = Q^* + 1$. And let the CEP after the q -th pass be $p(q)$ by the 2LP $\mathcal{P}_q(N^*)$. If

$$p(q) \geq \tilde{p} \cdot (1.05 + 0.1e^{-0.01q}) \quad (24)$$

then number t of trials to improve CEP is increased by 1. The improvement does not imply straightforward decrement of CEP, where the multiplier factor

$$\rho(q) = 1.05 + 0.1e^{-0.01q} \quad (25)$$

is just a relaxation function (Fig. 1). If $t = t_{\max}$ then the extra training is broken. Otherwise, if (24) is false then $\tilde{p} = p(q)$. Here, if $p(q) < p_*$ then $p_* = p(q)$. In the end, whatever \tilde{p} is, the best 2LP $\mathcal{P}_*(N^*)$ is saved (Fig. 2).

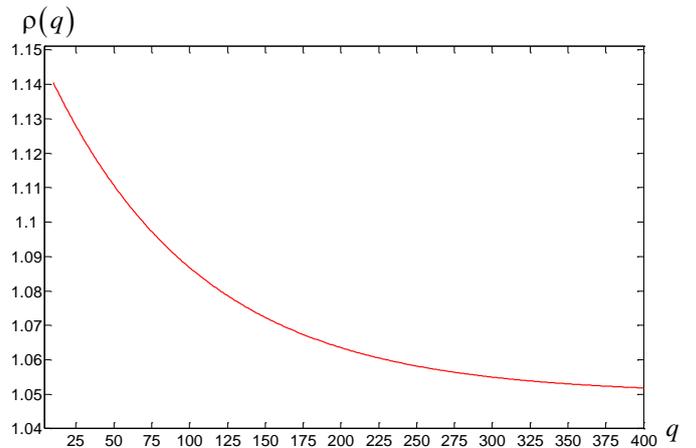


Fig. 1. The relaxation function (25) whose values are a multiplier factor to the temporary CEP. They show how the current CEP may increase so that this increment would be counted insignificant, and when inequality (24) is false we could accept 2LP $\mathcal{P}_q(N^*)$ after the q -th pass as the temporary one. The limit relaxation value is 5% that is about the testing inaccuracy.

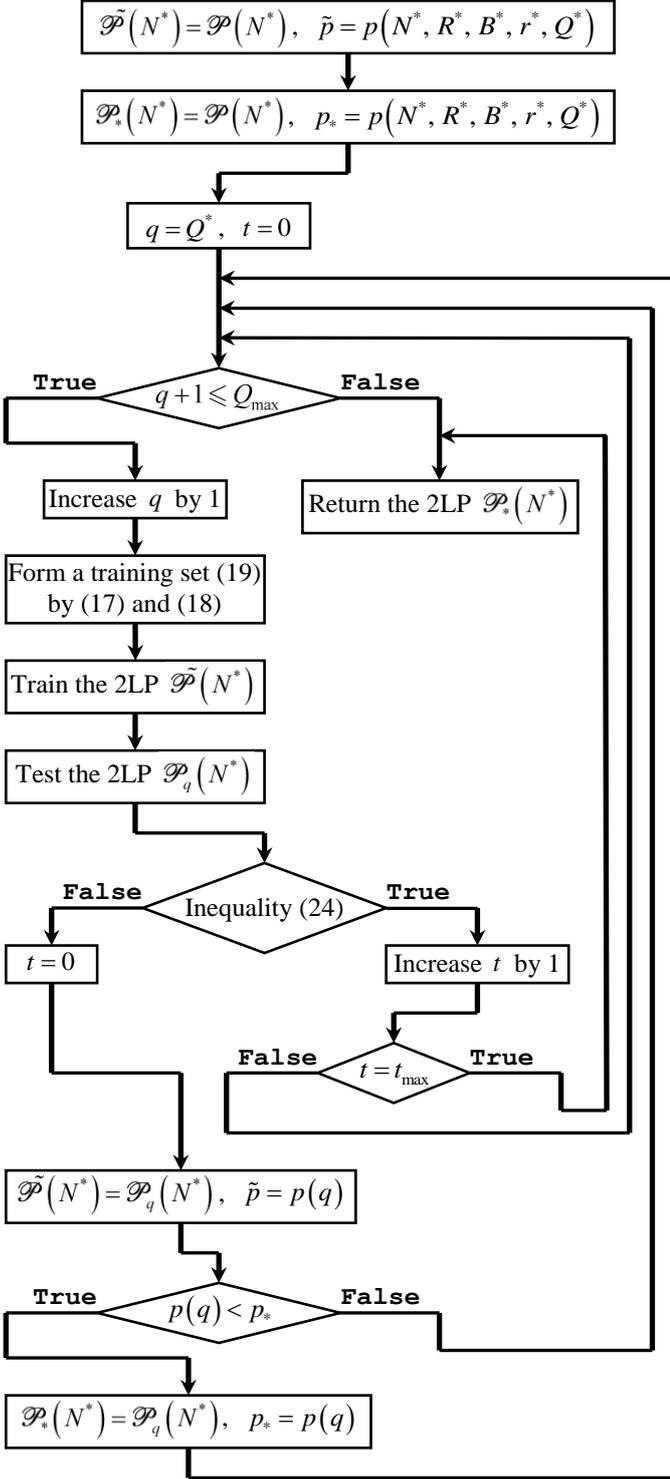


Fig. 2. The algorithm of training the 2LP with extra passes after Q^* passes. The temporary 2LP $\tilde{\mathcal{P}}(N^*)$ trained at the q -th pass becomes the 2LP $\mathcal{P}_q(N^*)$ after the training is accomplished. The algorithm uses the relaxation function (25) for inequality (24), but it is not limited to this function.

The extra pass training allows classifying 30×40 EACL by $C = 26$ at CEP 2.32 %. Unlike upper bound of CEP in (23), here, in the worst case, just one-fortieth EACL is classified wrong. Fig. 3 shows CEP for all 3×4 aspect ratio formats.

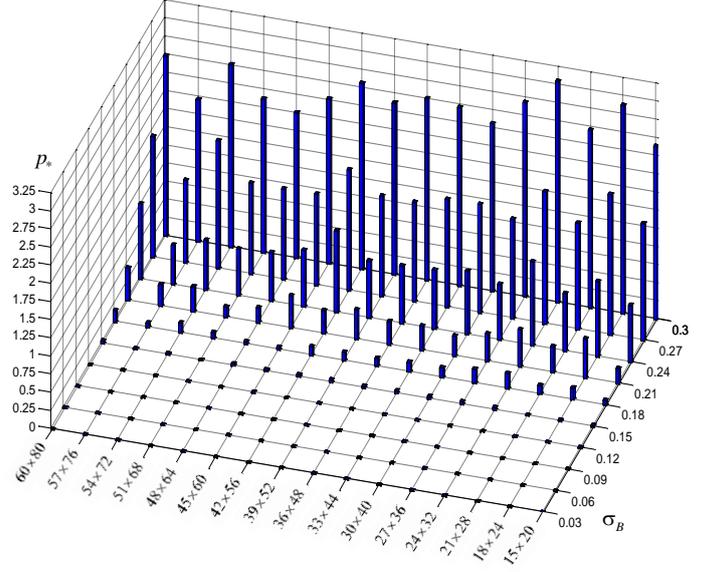


Fig. 3. CEP against SDI by SSD $\sigma_B \in \{0.03i\}_{i=1}^{10}$ for testing sets. Each 2LP classifier is tested after its extra pass training. PORs are classified without errors. DSSs are also classified without errors, if SDI is slight or moderate corresponding to SSD $\sigma_B < 0.12$, although by $\sigma_B = 0.15$ (a half of the SDI maximum) likelihood of an error is pretty insignificant. An outstanding notability is that CEP does not depend on number of features.

XIV. APPLICATION

The stated optimisation technique is easily applied to scaled object classification problems with casual aspect ratio formats. Moreover, the object does not need to have two dimensions [35], [36]. Application is suggested in that solution (22) is adjusted to the arisen problem. For this, HLNN is mostly tried [18], [37], [38]. For number $F > 1200$ it may be increased [14], especially when $C > 26$. And $N^* < 109$ if $F < 1200$ and $C < 26$. Subsequently, 2LP is trained with extra passes. The extra training should start right after the 47-th pass. The generalised relaxation function

$$\rho(q) = \rho_0 + \alpha e^{-\beta q}$$

$$\text{by } \rho_0 \in [1; 1.1) \text{ and } \alpha \in (0; 1) \text{ and } \beta \in (0; 1) \quad (26)$$

for the condition

$$p(q) < \tilde{p} \cdot \rho(q) \quad (27)$$

letting restart the extra pass control can be adjusted regarding specificities of the classification problem. Exponentially decreasing function (26) may be substituted for an appropriate decreasing curve, which is not necessary to be monotonous [39], [40].

XV. DISCUSSION

The polylines in Fig. 3 prove the 2LP is an effective classifier of the scaled objects. Although objects in the benchmark classification are of a few thousand features, there are no visible restrictions to use the extra trained 2LP in classifying scaled objects by any F . A demerit is the scaling distortion that has

been solely explored. Nevertheless, the results of the present research give evidence that the optimised 2LP is capable of handling other maximum distortion types. Solution (22) can be the starting point for them in searching their own respective solutions. Point (22) is anyway plausible to be the nearly closest to optimal training parameters and HLNN of 2LP aiming at classifying diversely distorted objects.

The extra passes prolong the training process greatly. For classifying 30×40 EACL by $C = 26$ and $t_{\max} = 150$ it takes up to 700 passes. The least 404 passes came on training the 54×72 EACL classifier. Training 15×20 EACL classifier took 409 passes. And training 60×80 EACL classifier takes no less than those 700 passes. Notwithstanding the dragging on cross-validation, the resulting 2LP performance is worth it. The best 2LP $\mathcal{P}_*(N^*)$ is expected to be readily tuned to a non-scaling distortion type problem by the same F . For implementing this, PSSDR is slightly tweaked and we put $\mathcal{P}(N^*) = \mathcal{P}_*(N^*)$ to start the extra training by the algorithm in Fig. 2.

XVI. CONCLUSION

The stated 2LP optimisation technique lies in minimising the CEP on 5DLL and extra pass training. Preliminarily, the CEP statistics in multidimensional matrices is accumulated. The CEP is evaluated by averaging over the dimension containing different 2LPs by the same HLNN and the identical training parameters. Then, HLNN and four parameters ensuring minimal CEP are chosen, and 2LP is trained. If the trained 2LP performance is not satisfactory, it is extra trained by the algorithm in Fig. 2.

Such an optimisation approach can be used for deep neural networks [41], [42]. The optimised 2LP is a potential base unit for deep learning, like for boosting [11], [16], [20], [43], [44]. The numerical optimisation approach is universal, and it admits any distortion types or types of objects. It is promising that both the training for accumulating CEP statistics and the testing shall be accelerated by GPU computations and parallelization [33], [34], [45], [46]. Very likely, using CUDA [47], [48] is preferable to higher-level languages.

Despite statements (5)–(12) in the model of scale distortion relate to flat objects, the third dimension is induced analogously. Training sets and testing sets (17)–(19) do not change. Hence, the scaled three-dimensional objects will be modelled and classified accurately as well. Problems of multidimensional scaling effects are going to be resolved similarly.

For generalised scaled object classification problem, it is expected that CEP for 3×4 aspect ratio format keeps similar. The barred graphs shown in Fig. 3 are generated the same every new testing time. At moderate SDI corresponding to $\text{SSD } \sigma_B = 0.1$, DSSs are classified without errors.

For developing the 2LP optimisation technique, a 2LP training method should be parameterised. For instance, the used backpropagation method updates weight and bias values according to gradient descent with adaptive learning rate [26], [27]. Its eight main internal parameters are maximum number

of epochs to train, performance goal, learning rate, ratio to increase learning rate, ratio to decrease learning rate, maximum validation failures, maximum performance increase, and minimum performance gradient. These ones are usually defined empirically, so their optimised aggregate would make the 2LP classification even more accurate.

ACKNOWLEDGEMENT

The present research has been technically supported by the Centre of Parallel Computations and its GPU clusters at Khmelnytsky National University, Ukraine.

REFERENCES

- [1] J. Fan, J. Zhang, K. Mei, J. Peng, and L. Gao, "Cost-sensitive learning of hierarchical tree classifiers for large-scale image classification and novel category detection," *Pattern Recognition*, vol. 48, iss. 5, 2015, pp. 1673–1687. <http://dx.doi.org/10.1016/j.patcog.2014.10.025>
- [2] J. You and H. A. Cohen, "Classification and segmentation of rotated and scaled textured images using texture "tuned" masks," *Pattern Recognition*, vol. 26, iss. 2, 1993, pp. 245–258. [http://dx.doi.org/10.1016/0031-3203\(93\)90033-S](http://dx.doi.org/10.1016/0031-3203(93)90033-S)
- [3] X. Wei, S. L. Phung, and A. Bouzerdoum, "Object segmentation and classification using 3-D range camera," *Journal of Visual Communication and Image Representation*, vol. 25, iss. 1, 2014, pp. 74–85. <http://dx.doi.org/10.1016/j.jvcir.2013.04.002>
- [4] W. Wei and Y. Xin, "Rapid, man-made object morphological segmentation for aerial images using a multi-scaled, geometric image analysis," *Image and Vision Computing*, vol. 28, iss. 4, 2010, pp. 626–633. <http://dx.doi.org/10.1016/j.imavis.2009.10.002>
- [5] J. Bai, Y. Wu, J. Zhang, and F. Chen, "Subset based deep learning for RGB-D object recognition," *Neurocomputing*, vol. 165, 2015, pp. 280–292. <http://dx.doi.org/10.1016/j.neucom.2015.03.017>
- [6] S. Kim, Y. Choi, and M. Lee, "Deep learning with support vector data description," *Neurocomputing*, vol. 165, 2015, pp. 111–117. <http://dx.doi.org/10.1016/j.neucom.2014.09.086>
- [7] K. Polat and S. Güneş, "A novel hybrid intelligent method based on C4.5 decision tree classifier and one-against-all approach for multi-class classification problems," *Expert Systems with Applications*, vol. 36, iss. 2, part 1, 2009, pp. 1587–1592. <http://dx.doi.org/10.1016/j.eswa.2007.11.051>
- [8] D. M. Farid, L. Zhang, C. M. Rahman, M. A. Hossain, and R. Strachan, "Hybrid decision tree and naïve Bayes classifiers for multi-class classification tasks," *Expert Systems with Applications*, vol. 41, iss. 4, part 2, 2014, pp. 1937–1946. <http://dx.doi.org/10.1016/j.eswa.2013.08.089>
- [9] A. Özçift, "Random forests ensemble classifier trained with data resampling strategy to improve cardiac arrhythmia diagnosis," *Computers in Biology and Medicine*, vol. 41, iss. 5, 2011, pp. 265–271. <http://dx.doi.org/10.1016/j.combiomed.2011.03.001>
- [10] A. Fernández-Baldera and L. Baumela, "Multi-class boosting with asymmetric binary weak-learners," *Pattern Recognition*, vol. 47, iss. 5, 2014, pp. 2080–2090. <http://dx.doi.org/10.1016/j.patcog.2013.11.024>
- [11] V. V. Romanuke, "Boosting ensembles of heavy two-layer perceptrons for increasing classification accuracy in recognizing shifted-turned-scaled flat images with binary features," *Journal of Information and Organizational Sciences*, vol. 39, no. 1, 2015, pp. 75–84.
- [12] V. V. Romanuke, "Pixel-to-scale standard deviations ratio optimization for two-layer perceptron training on pixel-distorted scaled 60-by-80-images in scaled objects classification problem," *Transactions of Kremenchuk Mykhailo Ostrohradskyi National University*, iss. 2 (85), 2014, pp. 96–105.
- [13] H. Shao, J. Wang, L. Liu, D. Xu, and W. Bao, "Relaxed conditions for convergence of batch BPAP for feedforward neural networks," *Neurocomputing*, vol. 153, 2015, pp. 174–179. <http://dx.doi.org/10.1016/j.neucom.2014.11.039>
- [14] F. Cao, S. Lin, and Z. Xu, "Approximation capability of interpolation neural networks," *Neurocomputing*, vol. 74, iss. 1–3, 2010, pp. 457–460. <http://dx.doi.org/10.1016/j.neucom.2010.08.018>
- [15] V. V. Romanuke, "An attempt for 2-layer perceptron high performance in classifying shifted monochrome 60-by-80-images via training with pixel-distorted shifted images on the pattern of 26 alphabet letters,"

- Radioelectronics, informatics, control*, no. 2, 2013, pp. 112–118. <http://dx.doi.org/10.15588/1607-3274-2013-2-18>
- [16] V. V. Romanuke, "Accuracy improvement in wear state discontinuous tracking model regarding statistical data inaccuracies and shifts with boosting mini-ensemble of two-layer perceptrons," *Problems of tribology*, no. 4, 2014, pp. 55–58.
- [17] P. A. Castillo, J. J. Merelo, M. G. Arenas, and G. Romero, "Comparing evolutionary hybrid systems for design and optimization of multilayer perceptron structure along training parameters," *Information Sciences*, vol. 177, iss. 14, 2007, pp. 2884–2905. <http://dx.doi.org/10.1016/j.ins.2007.02.021>
- [18] V. V. Romanuke, "Setting the hidden layer neuron number in feedforward neural network for an image recognition problem under Gaussian noise of distortion," *Computer and Information Science*, vol. 6, no. 2, 2013, pp. 38–54. <http://dx.doi.org/10.5539/cis.v6n2p38>
- [19] G. Arulampalam and A. Bouzerdoum, "A generalized feedforward neural network architecture for classification and regression," *Neural Networks*, vol. 16, iss. 5–6, 2003, pp. 561–568. [http://dx.doi.org/10.1016/S0893-6080\(03\)00116-3](http://dx.doi.org/10.1016/S0893-6080(03)00116-3)
- [20] V. V. Romanuke, "Optimizing parameters of the two-layer perceptrons' boosting ensemble training for accuracy improvement in wear state discontinuous tracking model regarding statistical data inaccuracies and shifts," *Problems of tribology*, no. 1, 2015, pp. 65–68.
- [21] P. G. Benardos and G.-C. Vosniakos, "Optimizing feedforward artificial neural network architecture," *Engineering Applications of Artificial Intelligence*, vol. 20, iss. 3, 2007, pp. 365–382. <http://dx.doi.org/10.1016/j.engappai.2006.06.005>
- [22] V. V. Romanuke, "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"*, no. 5, 2014, pp. 55–62.
- [23] V. V. Romanuke, "Classification error percentage decrement of two-layer perceptron for classifying scaled objects on the pattern of monochrome 60-by-80-images of 26 alphabet letters by training with pixel-distorted scaled images," *Scientific bulletin of Chernivtsi National University of Yuriy Fedkovych. Series: Computer systems and components*, vol. 4, iss. 3, 2013, pp. 53–64.
- [24] G. A. Anastassiou, "Multivariate sigmoidal neural network approximation," *Neural Networks*, vol. 24, iss. 4, 2011, pp. 378–386. <http://dx.doi.org/10.1016/j.neunet.2011.01.003>
- [25] D. Costarelli and R. Spigler, "Approximation results for neural network operators activated by sigmoidal functions," *Neural Networks*, vol. 44, 2013, pp. 101–106. <http://dx.doi.org/10.1016/j.neunet.2013.03.015>
- [26] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, iss. 6, 1994, pp. 989–993. <http://dx.doi.org/10.1109/72.329697>
- [27] T. Kathirvalavakumar and S. Jeyaseeli Subavathi, "Neighborhood based modified backpropagation algorithm using adaptive learning parameters for training feedforward neural networks," *Neurocomputing*, vol. 72, iss. 16–18, 2009, pp. 3915–3921. <http://dx.doi.org/10.1016/j.neucom.2009.04.010>
- [28] K. Hagiwara, T. Hayasaka, N. Toda, S. Usui, and K. Kuno, "Upper bound of the expected training error of neural network regression for a Gaussian noise sequence," *Neural Networks*, vol. 14, iss. 10, 2001, pp. 1419–1429. [http://dx.doi.org/10.1016/S0893-6080\(01\)00122-8](http://dx.doi.org/10.1016/S0893-6080(01)00122-8)
- [29] Z. Chen, F. Cao, and J. Hu, "Approximation by network operators with logistic activation functions," *Applied Mathematics and Computation*, vol. 256, 2015, pp. 565–571. <http://dx.doi.org/10.1016/j.amc.2015.01.049>
- [30] J. M. Alcaraz Calero and J. G. Aguado, "Comparative analysis of architectures for monitoring cloud computing infrastructures," *Future Generation Computer Systems*, vol. 47, 2015, pp. 16–30. <http://dx.doi.org/10.1016/j.future.2014.12.008>
- [31] T. Püschel, G. Schryen, D. Hristova, and D. Neumann, "Revenue management for Cloud computing providers: Decision models for service admission control under non-probabilistic uncertainty," *European Journal of Operational Research*, vol. 244, iss. 2, 2015, pp. 637–647. <http://dx.doi.org/10.1016/j.ejor.2015.01.027>
- [32] A. D. Kshemkalyani and M. Singhal, *Distributed Computing Principles, Algorithms, and Systems*. Cambridge: Cambridge University Press, 2008.
- [33] R. Trobec, M. Vajteršic, and P. Zinterhof, Eds., *Parallel Computing, Numerics, Applications, and Trends*. London: Springer, 2009.
- [34] W. P. Petersen and P. Arbenz, *Introduction to Parallel Computing: A practical guide with examples in C*. Oxford: Oxford University Press, 2004.
- [35] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision. Third Edition*. Toronto: Thomson, 2008.
- [36] D. A. Forsyth and J. Ponce, *Computer Vision. A Modern Approach. Second Edition*. Upper Saddle River, NJ: Pearson, 2012.
- [37] V. E. Ismailov, "On the approximation by neural networks with bounded number of neurons in hidden layers," *Journal of Mathematical Analysis and Applications*, vol. 417, iss. 2, 2014, pp. 963–969. <http://dx.doi.org/10.1016/j.jmaa.2014.03.092>
- [38] C. Yu, M. T. Manry, J. Li, and P. L. Narasimha, "An efficient hidden layer training method for the multilayer perceptron," *Neurocomputing*, vol. 70, iss. 1–3, 2006, pp. 525–535. <http://dx.doi.org/10.1016/j.neucom.2005.11.008>
- [39] V. V. Romanuke, "Theoretic-game methods of identification of models for multistage technical control and run-in under multivariate uncertainties," a dissertation for the doctoral degree of technical sciences in speciality 01.05.02—mathematical modeling and computational methods, Vinnytsia National Technical University, Vinnytsia, Ukraine, 2014 (in Ukrainian).
- [40] V. V. Romanuke, "Convergence and estimation of the process of computer implementation of the optimality principle in matrix games with apparent play horizon," *Journal of Automation and Information Sciences*, vol. 45, iss. 10, 2013, pp. 49–56. <http://dx.doi.org/10.1615/JAutomatInfScien.v45.i10.70>
- [41] S. M. Siniscalchi, D. Yu, L. Deng, and C.-H. Lee, "Exploiting deep neural networks for detection-based speech recognition," *Neurocomputing*, vol. 106, 2013, pp. 148–157. <http://dx.doi.org/10.1016/j.neucom.2012.11.008>
- [42] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, 2015, pp. 85–117. <http://dx.doi.org/10.1016/j.neunet.2014.09.003>
- [43] Q. Nie, L. Jin, and S. Fei, "Probability estimation for multi-class classification using AdaBoost," *Pattern Recognition*, vol. 47, iss. 12, 2014, pp. 3931–3940. <http://dx.doi.org/10.1016/j.patcog.2014.06.008>
- [44] S. Zheng, "QBoost: Predicting quantiles with boosting for regression and binary classification," *Expert Systems with Applications*, vol. 39, iss. 2, 2012, pp. 1687–1697. <http://dx.doi.org/10.1016/j.eswa.2011.06.060>
- [45] F. Silber-Chaussumier, A. Muller, and R. Habel, "Generating data transfers for distributed GPU parallel programs," *Journal of Parallel and Distributed Computing*, vol. 73, iss. 12, 2013, pp. 1649–1660. <http://dx.doi.org/10.1016/j.jpdc.2013.07.022>
- [46] J. Lee, J.-H. Park, H. Kim, C. Jung, D. Lim, and S. Y. Han, "Adaptive execution techniques of parallel programs for multiprocessors," *Journal of Parallel and Distributed Computing*, vol. 70, iss. 5, 2010, pp. 467–480. <http://dx.doi.org/10.1016/j.jpdc.2009.10.008>
- [47] C. Obrecht, F. Kuznik, B. Tourancheau, and J.-J. Roux, "Scalable lattice Boltzmann solvers for CUDA GPU clusters," *Parallel Computing*, vol. 39, iss. 6–7, 2013, pp. 259–270. <http://dx.doi.org/10.1016/j.parco.2013.04.001>
- [48] M. Krotkiewski and M. Dabrowski, "Efficient 3D stencil computations using CUDA," *Parallel Computing*, vol. 39, iss. 10, 2013, pp. 533–548. <http://dx.doi.org/10.1016/j.parco.2013.08.002>

Vadim Romanuke graduated from the Technological University of Podillya (Ukraine) in 2001. In 2006, he received the degree of Candidate of Technical Sciences in Mathematical Modelling and Computational Methods. The degree of Doctor of Technical Sciences in Mathematical Modelling and Computational Methods was received in 2014.

He is a Professor of the Department of Applied Mathematics and Social Informatics at Khmelnytsky National University. His current research interests concern decision making, game theory, statistical approximation, and control engineering based on statistical correspondence. He has 278 published scientific articles and one tutorial.

Annually, he is the Head either of sections of (applied) mathematics or mathematical modelling in mathematics branch for regional school scientific competitions. He is regularly awarded by Khmelnytsky National University for scientific achievements.

Address for correspondence: 11 Institut'skaya Str., 29016, Khmelnytsky, Ukraine.

E-mail: romanukevadimv@mail.ru