Neural Comput & Applic
DOI 10.1007/s00521-016-2754-1

ORIGINAL ARTICLE

Extreme learning machine for structured output spaces

Ayman Maliha[1] · **Rubiyah Yusof**[1] · **Mohd Ibrahim Shapiai**[1]

Received: 15 March 2016 / Accepted: 23 November 2016

Abstract Recently, extreme learning machine (ELM) has attracted increasing attention due to its successful applications in classification, regression, and ranking. Normally, the desired output of the learning system using these machine learning techniques is a simple scalar output. However, there are many applications in machine learning which require more complex output rather than a simple scalar one. Therefore, structured output is used for such applications where the system is trained to predict structured output instead of simple one. Previously, support vector machine (SVM) has been introduced for structured output learning in various applications. However, from machine learning point of view, ELM is known to offer better generalization performance compared to other learning techniques. In this study, we extend ELM to more generalized framework to handle complex outputs where simple outputs are considered as special cases of it. Besides the good generalization property of ELM, the resulting model will possesses rich internal structure that reflects task-specific relations and constraints. The experimental results show that structured ELM achieves similar (for binary problems) or better (for multi-class problems) generalization performance when compared to ELM. Moreover, as verified by the simulation results, structured ELM has comparable or better precision performance with structured SVM when tested for more complex output such as object localization problem on PASCAL VOC2006. Also, the investigation on parameter selections is presented and discussed for all problems.

Keywords Extreme learning machine (ELM) · Structured learning · Object detection · Quadratic programming

✉ Ayman Maliha
amaliha@iugaza.edu.ps

Rubiyah Yusof
rubiyah.kl@utm.my

Mohd Ibrahim Shapiai
ibrahimfke@gmail.com

[1] Centre for Artificial Intelligence and Robotics, Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia

1 Introduction

In recent years, single-hidden-layer feedforward neural networks (SLFNs) has been of interest to many researchers due to its ability to approximate complex nonlinear mapping directly from input samples [1–3]. ELM was originally developed for the SLFNs [3–5] and then extended to generalized SLFNs which may not be a neuron like [6, 7]. ELM has been shown to be extremely fast in training phase, so it is ranked high for fast implementation among machine learning methods [8–10]. Also, ELM has a number of good features such as simple in theory, variation of hidden nodes feature mapping or kernel form, and better generalization performance than other supervised learning methods [4, 11–14].

In general, different forms of ELM have been proposed in the literature. The basic form of ELM has been proposed in [3, 4] where the learning parameters between the input nodes and hidden nodes are generated randomly and do not need to be tuned. Then, the training is completed by analytically finding the output nodes weight using least square solution. The kernel-based ELM is studied in [9] where the feature mapping and the dimensionality of the hidden layer need not to be known to users and instead a kernel is given. A kernel implementation when the feature mapping is

Published online: 10 December 2016

known is studied in [15, 16]. To provide good prediction and fast learning, kernel ELM was applied in damage location detection as in [17]. Online sequential ELM (OS-ELM) has been proposed in [10] to enable adaptive learning. The learning algorithm accepts the training data sequentially as chunks and discards the chunk for which the training has already been done. In incremental learning method, the hidden layer nodes are generated randomly and added to the existing network until the number of hidden nodes exceeds a predefined maximum number or the residual error becomes less than the expected one [5–7]. In basic incremental ELM (I-ELM) [5, 6], the hidden nodes are added one by one to the existing network, whereas in enhanced I-ELM [7], a number of hidden nodes are generated, and then the most appropriate nodes will be added to the network. A pruned ELM (P-ELM) was presented in [18] for a classification problem, where hidden nodes with low relevance to class labels are eliminated from a large network. ELM ensemble was proposed in [19] to predict sales amount, where several ELM networks were integrated in parallel. The final output was calculated as the average of the ELMs' outputs and resulted in better generalization performance. Relevance ranking or learning to rank is a popular topic in many areas such as sentiment analysis, document retrieval [20–22]. ELM with linear random node was applied to relevance ranking in [20], where pointwise RankELM and pairwise RankELM were proposed and tested on publicly available data set collection.

Research areas and applications for computer vision, natural language processing, protein structure prediction, and information retrieval require highly complex and accurate models [23]. Recent developments to deal with complex models have been investigated in order to satisfy additional constraints, i.e., the output has structure [24]. Problems with structured output spaces such as object detection in moving or still images, sequences, labeled trees, strings, lattices, or graphs have shown to be solved as structured prediction models [23, 25, 26]. Furthermore, the binary and multi-class classification problems can be solved using structured models where they are considered as special cases of it [23].

A generalization of SVM for structured output has been proposed in [27], where the loss function is rescaled using two methods, slack rescaling and margin rescaling. Structured output is also used in computer vision application such as object localization [25, 28] and object tracking [26]. In [25], the structured output space consists of a label, indicating whether an object is present, and a vector indicating the top, left, bottom, and right coordinates of the bounding box within the image. The mapping function is learnt in structured learning framework. In [26], a kernelized structured output with SVM was used as online learning scheme to train an adaptive tracking system.

ELM and SVM are equivalent from standard optimization method point of view, but ELM has less optimization constraints [8]. Furthermore, ELM usually has feature mapping that is known to user and adopts randomness. Almost all nonlinear piecewise continuous functions can be used as feature mappings. Also if the feature mapping is not known to user, kernels can be applied in ELM similar to SVM [8, 9]. From machine learning point of view, ELM actually provides a unified framework where the generalized decision from training data solve binary classification or predicting a scalar number as in the regression. In general, ELM can achieve similar or better generalization performance than SVM in regression and binary classification problems. Moreover, ELM has much better generalization performance than SVM in multi-class problems [8, 9].

Simple output used in classification and regression may not be a good solution for highly complex and accurate models required in the aforementioned research areas. The required output of these models normally consists of structured output rather than simple one [27]. In this paper, we propose a new ELM model that is trained to predict structured or complex output. Also, the proposed model will treat the binary and multi-class classification problems as structural learning problems. To the best of our knowledge, there is no work of applying ELM with structured output so far.

The rest of the paper is organized as follows. In the next section, we review primal ELM architecture model and its learning algorithm. In Sect. 3, we introduce the proposed structured output learning for ELM. The experimental setting and results discussion are presented in Sect. 4 for three different applications. Finally, a conclusion is drawn in Sect. 5.

2 Extreme learning machine

ELM model is based on SLFNs architecture model [29]. It consists of three layers: the input layer to receive the stimuli from the external environments, the hidden layer, and the output layer to send network output to the external environments. Although ELM output layer can contain more than one node, in this paper, we will adopt a single output node as shown in Fig. 1.

Not like conventional machine learning approaches, ELM requires less parameter selection and can be fast in training with good generalization performance. Many properties make ELM suitable to be applied in many research areas and fields including real-time and online applications. Such properties are (1) easiness of use, (2) fast learning speed, (3) high generalization performance, (4) suitable for many nonlinear activation functions and kernel functions, and (5) batch and sequential learning [10].

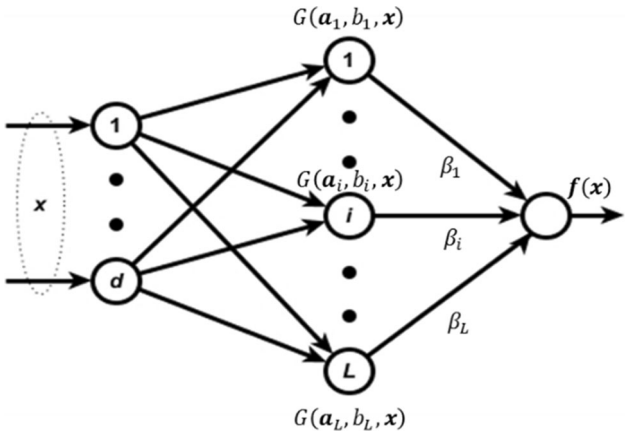

Fig. 1 ELM model with single output node

The output of ELM, with single output node and L hidden nodes is mathematically formulated by:

$$f(\boldsymbol{x}) = \sum_{i=1}^{L} \beta_i G(\boldsymbol{a_i}, b_i, \boldsymbol{x}) \tag{1}$$

where $\boldsymbol{a_i}$ are input weights from d input nodes to ith hidden node and b_i is the bias of the ith hidden node. $\boldsymbol{a_i}$ and b_i are called the learning parameters of the ith hidden node. $\boldsymbol{x}$ is the input vector with d dimensions, and β_i is the output weight from the ith hidden node. $G(\boldsymbol{a_i}, b_i, \boldsymbol{x})$ is the output of the ith hidden node with respect to input $\boldsymbol{x}$ and G activation function. Activation function can be any nonlinear piecewise continuous functions such as sigmoid, Gaussian.

In a training phase, both hidden nodes and the output node parameters are needed to be determined. In ELM theory, the hidden node parameters $\boldsymbol{a_i}$ and b_i are assigned values randomly regardless of the nature of the input and remained fixed after that. In this case, β_i is the only parameter needs to be determined using the training data.

In ELM, given a training data $(\boldsymbol{x}_i, t_i)$, $i = 1, \ldots, N$, where $\boldsymbol{x}_i \in \boldsymbol{R}^d$ and $t_i \in \{-1, +1\}$, we want to minimize the training error in the cost function that is formed in least square sense and given by Eq. (2).

$$E = \min \sum_{j=1}^{N} \left(\sum_{i=1}^{L} \beta_i G(\boldsymbol{a}_i, b_i, \boldsymbol{x}_j) - t_j \right)^2 \tag{2}$$

Equation 2 can be written in a compact form as in (3)

$$\boldsymbol{H\beta} = \boldsymbol{T} \tag{3}$$

where $\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{h}(\boldsymbol{x}_N) \end{bmatrix} = \begin{bmatrix} G(\boldsymbol{a}_1, b_1, \boldsymbol{x}_1) & \ldots & G(\boldsymbol{a_L}, b_L, \boldsymbol{x}_1) \\ \vdots & \ldots & \vdots \\ G(\boldsymbol{a}_1, b_1, \boldsymbol{x}_N) & \ldots & G(\boldsymbol{a_L}, b_L, \boldsymbol{x}_N) \end{bmatrix}_{N \times L}$,

$\boldsymbol{\beta} = \begin{bmatrix} \beta_1 \\ \vdots \\ \beta_L \end{bmatrix}_{L \times 1}$, and $\boldsymbol{T} = \begin{bmatrix} t_1 \\ \vdots \\ t_N \end{bmatrix}_{N \times 1}$. $\boldsymbol{H}$ is the hidden layer output matrix of the SLFN [13, 26]. The ith column of $\boldsymbol{H}$ is the ith hidden node output with respect to inputs $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N$. $\boldsymbol{h}(\boldsymbol{x})$ is defined as $\boldsymbol{h}(\boldsymbol{x}) = [G(\boldsymbol{a}_1, b_1, \boldsymbol{x}), \ldots, G(\boldsymbol{a_L}, b_L, \boldsymbol{x})]$ and called the hidden layer feature mapping [30, 31]. The ith row of $\boldsymbol{H}$ is the hidden layer feature mapping with respect to the ith input $\boldsymbol{x}_i$. So, solving the linear system in Eq. (3) for $\boldsymbol{\beta}$ is equivalent to training the network. If the number of training samples is equal to the number of hidden nodes, $L = N$, then $\boldsymbol{H}$ is a square matrix and $\boldsymbol{\beta}$ can be found by calculating the inverse of $\boldsymbol{H}$ where a zero training error is obtained in this case. If $L < N$, it is not a square matrix and a solution can be found using Moore–Penrose generalized inverse of matrix $\boldsymbol{H}$ as given in Eq. (4) [32].

$$\boldsymbol{\beta} = \boldsymbol{H}^{\dagger} \boldsymbol{T} \tag{4}$$

In binary classification problems, the decision function for ELM with one output node can be written in a vector form from Eq. (1) and is given in Eq. (5).

$$f(\boldsymbol{x}) = \operatorname{sign}\left(\sum_{i=1}^{L} \beta_i G(\boldsymbol{a_i}, b_i, \boldsymbol{x}) \right) = \operatorname{sign}(\boldsymbol{\beta} \cdot \boldsymbol{h}(\boldsymbol{x})) \tag{5}$$

where, $\boldsymbol{\beta}$ is the estimated output weight vector in Eq. (4) and $\boldsymbol{h}(\boldsymbol{x})$ is a vector that maps the d-dimensional input space to L-dimensional hidden layer feature space [8].

3 Structured output learning for ELM

Structured learning is a learning method to predict complex output (structured prediction) rather than simple classification or regression which normally includes simple scalar outputs. Structured prediction handles compound or structured variables such as sequences, strings, trees, or graphs. The important key of structured learning is the inclusion of the interdependencies between outputs in the learning stage in addition to the dependency that exist between inputs and outputs. The interdependencies can be formulated as constraints that restrict the possible outputs [24, 25]. For most practical problems, the output is an arbitrary object that can be represented by a discrete vector $y = (y_1, y_2, \ldots, y_k)$. For example, in object localization problem, the output space could be the entire bounding boxes in an image. The bounding boxes can be represented, for example, by four variables indicating the top, left, right, and bottom coordinates of the region, i.e., $y =$ (top, left, right, bottom). The variables in this case can be assigned values between 0 and the image size as shown in Fig. 2.

In ELM training, ELM tends to reach the smallest training error as well as the smallest norm of the output

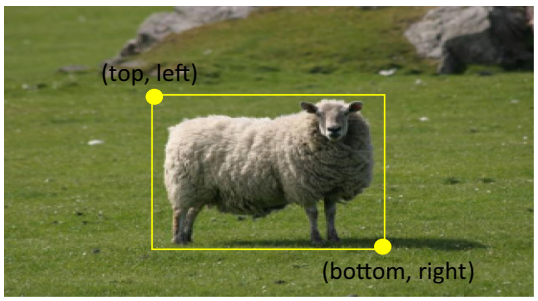

Fig. 2 Object localization using *top*, *left*, *bottom*, and *right* coordinates of a bounding *box*

weights [3, 4], so the problem can be formulated as given in Eq. (6):

$$\text{minimize:} \quad \sum_{i=1}^{N} \|\boldsymbol{\beta} \cdot h(\boldsymbol{x}_i) - t_i\| \quad \text{and} \quad \text{minimize} \|\boldsymbol{\beta}\| \tag{6}$$

In ELM, minimizing the norm of output weights $\|\boldsymbol{\beta}\|$ is actually maximizing the distance of the separating margins of two different classes in ELM feature space: $2/\|\boldsymbol{\beta}\|$. In standard optimization theory, the objective function in (6) for minimizing both the training error and the output norm for ELM can be written as in Eq. (7)

$$\begin{aligned} &\text{minimize:} \quad L_P = \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\sum_{i=1}^{N}\xi_i \\ &\text{subject to:} \quad t_i\boldsymbol{\beta} \cdot \boldsymbol{h}(\boldsymbol{x}_i) \geq 1 - \xi_i, \quad i = 1, \ldots, N \\ &\xi_i \geq 0, \quad i = 1, \ldots, N \end{aligned} \tag{7}$$

In structured prediction, a structured output $y \in \mathcal{Y}$ is predicted from input data $\boldsymbol{x} \in \mathcal{X}$ using a prediction function $f : \mathcal{X} \to \mathcal{Y}$. $\mathcal{Y}$ is a discrete structured output space where element $y \in \mathcal{Y}$ can be sequences, labeled trees, graphs, or bounding box of object in an image. For a given instance $\boldsymbol{x} \in \mathcal{X}$, the prediction function $f(\boldsymbol{x})$ is obtained by maximizing an auxiliary evaluation function $g(\boldsymbol{x}, y)$ over all possible elements in $\mathcal{Y}$. The auxiliary function is trained to evaluate the quality of each possible structured output for a given input $\boldsymbol{x}$. In the same manner, the structure prediction function for ELM framework can be defined as given in Eq. (8).

$$f(\boldsymbol{x}) = \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, g(\boldsymbol{x}, y) \tag{8}$$

where $g(\boldsymbol{x}, y)$ is the output function of ELM network and defined as $g(\boldsymbol{x}, y) = \langle \boldsymbol{\beta}, \boldsymbol{h}(\boldsymbol{x}, y) \rangle$. This function scores an instance $\boldsymbol{x}$ for a given structured output y, in other words $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$. Hence, the task now is to learn g as a discriminant function in the given form of Eq. (8), i.e., to determine the values of the output weights $\boldsymbol{\beta}$ in ELM network. Standard optimization form of ELM is used in structural learning framework to learn the output weights of ELM network for structured outputs, and it is given in Eq. (9).

$$\begin{aligned} &\min_{\boldsymbol{\beta}, \xi} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i \\ &\text{subject to:} \quad \xi_i \geq 0, \quad \forall i \\ &\langle \boldsymbol{\beta}, \boldsymbol{h}(\boldsymbol{x}_i, y_i) \rangle - \langle \boldsymbol{\beta}, \boldsymbol{h}(\boldsymbol{x}_i, y) \rangle \geq \Delta(y_i, y) - \xi_i, \quad \forall i, \forall y \in \mathcal{Y} \backslash y_i \end{aligned} \tag{9}$$

where $\boldsymbol{h}(\boldsymbol{x}_i, y_i)$ is the joint kernel map and defined as the output of the hidden layer nodes in ELM network. The evaluation function given by $g(\boldsymbol{x}_i, y) = \langle \boldsymbol{\beta}, \boldsymbol{h}(\boldsymbol{x}_i, y) \rangle$ is the output of ELM network. The optimization in Eq. (9) aims to ensure that the output value of ELM network $g(\boldsymbol{x}_i, y_i)$ for a given instance $\boldsymbol{x}_i$ with its correct structured output y_i is larger than the output value with any other incorrect structured outputs, i.e., $g(\boldsymbol{x}_i, y)$ for $y \neq y_i$, by a margin that depends on the loss function $\Delta(y_i, y)$. The loss function, $\Delta(y_i, y)$, plays an important role in structured learning, and it is application dependent. For instance, the loss function for object localization problem is based on the overlapping of the correct output (bounding box) to other possible outputs in the training stage. For more details, please refer to Sect. 4. Using the example of object localization, we have the input space $\mathcal{X} = \{\text{images}\}$ and the output space $\mathcal{Y} = \mathbb{R}^4$ which represents the four coordinates of different bounding boxes; the prediction function given in Eq. (8) is used to find the maximum score for the correct prediction as shown in Fig. 3, where y_1 has the highest score.

The optimization in Eq. (9) is convex quadratic program (QP) with large number of constraints, e.g., the number of training samples times the size of the output space, or it becomes even infinite when the output is continuous. There are many solutions that have been proposed to solve this kind of optimization [23, 33–35]. The common approach of these methods is to decompose the training problem into smaller QPs that are to be solved. In [23], a cutting plane method with "1-slack" reformulation is used to obtain lower time complexity that is linear to number of training samples. The key idea of this is to use a single cutting plane model for the sum of hinge losses instead of n cutting plane of the hinge loss. Therefore, Eq. (9) can be written as in Eq. (10)

$$\begin{aligned} &\min_{\boldsymbol{\beta}, \xi \geq 0} \quad \frac{1}{2}\|\boldsymbol{\beta}\|^2 + C\xi \\ &\text{subject to:} \quad \forall(\bar{y}_1, \ldots, \bar{y}_N) \in \mathcal{Y}^N \\ &\frac{1}{N}\sum_{i=1}^{N}[\langle \boldsymbol{\beta}, \boldsymbol{h}(\boldsymbol{x}_i, y_i) \rangle - \langle \boldsymbol{\beta}, \boldsymbol{h}(\boldsymbol{x}_i, \bar{y}_i) \rangle] \geq \frac{1}{N}\sum_{i=1}^{N}\Delta(y_i, \bar{y}_i) - \xi \end{aligned} \tag{10}$$

where $\mathcal{Y}^N$ is the set of all possible combination of structures $\bar{y}$. The algorithm constructs a working set of

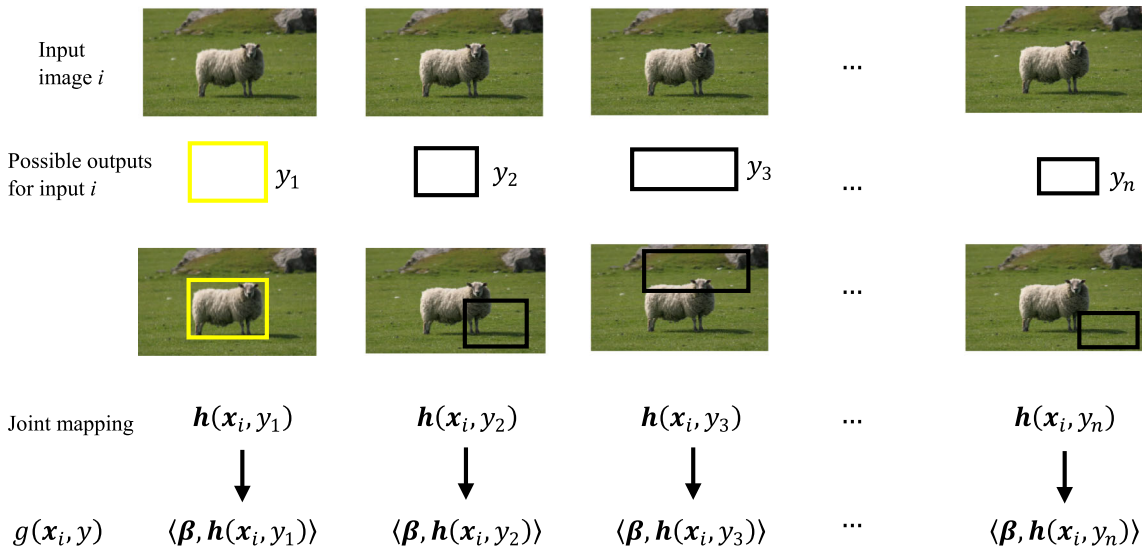

Fig. 3 Finding the highest score using Eq. (8)

constraints, $\mathcal{W}$ iteratively. Each constraint is basically based on the choice of the structured output. In each iteration, the solution over the current working set $\mathcal{W}$ is computed and then the most violated constraint is found and added to the working set to be used in the solution of the next iteration. In margin rescaling method, Eq. (11) is used to find the most violated constraint for the ith training example $(\boldsymbol{x}_i, y_i)$, where the structured output is selected when the value of the expression in (11) is the maximum for all $\hat{y} \in \mathcal{Y}$.

$$\hat{y}_i \leftarrow \underset{\hat{y} \in \mathcal{Y}}{\operatorname{argmax}} \{ \Delta(y_i, \hat{y}) - \langle \boldsymbol{\beta}, \boldsymbol{h}(\boldsymbol{x}_i, \hat{y}) \rangle \} \tag{11}$$

The algorithm continues in iterations and adding more constraints until no constraint can be found that is violated by more than a desired precision ε, for more details please refer to [23].

4 Experiments, results and discussion

Structured learning is a generalized learning framework that can be applied to different applications that are ranging from simple outputs which are special cases to complex structural outputs. In this paper, we will consider the following three applications, namely binary classification, multi-class classification, and object detection problem for more complex output. In binary and multi-class classification problems, we benchmark structured ELM with ELM itself since it is verified to achieve similar (for binary class cases) or much better (for multi-class cases) generalization performance than SVM [9]. In object localization cases, the proposed method is compared to structured SVM. For each application, a particular setting is required to define both the joint mapping and the loss functions.

Binary classifications For binary classification, we have $\mathcal{X} = R^d$ and $\mathcal{Y} = \{-1, 1\}$, and the joint mapping and the loss function are defined as follow:

$$\boldsymbol{h}_{\text{binary}}(\boldsymbol{x}, y) = \frac{y}{2} \boldsymbol{h}(\boldsymbol{x}, y) \quad \text{and} \quad \Delta(y_i, y) = \begin{cases} 0 & \text{if } y_i = y \\ 1 & \text{otherwise} \end{cases} \tag{12}$$

which is similar to binary classification problem of ELM in standard optimization form but in structural learning framework. Structural learning algorithm is used for optimization to find the weights of the output node.

Multi-class classification This is another example of structural learning where Kesler construction is used to build the joint mapping function [23]. In this case, we have $\mathcal{X} = R^d$ and $\mathcal{Y} = \{1, \ldots, K\}$, and the loss function is defined in a similar way like binary classification, and the joint mapping function is given by the stacked input, $\boldsymbol{x}_m$, where the feature vector $\boldsymbol{x}$ is stacked into position y of $\boldsymbol{x}_m$. The dimension of $\boldsymbol{x}_m$ equals to the number of classes times the number of features in $\boldsymbol{x}$, i.e., $K \times d$ where the joint mapping and loss are given in Eq. (13).

$$\boldsymbol{h}_{\text{multi}}(\boldsymbol{x}, y) = \boldsymbol{h}(\boldsymbol{x}_m, y) \quad \text{and} \quad \Delta(y_i, y) = \begin{cases} 0 & \text{if } y_i = y \\ 1 & \text{otherwise} \end{cases} \tag{13}$$

where $\boldsymbol{x}_m = [\, 0 \quad \ldots \quad 0 \quad \boldsymbol{x}^T \quad 0 \quad \ldots \quad 0 \,]^{\mathrm{T}}$. The argmax for both the separation oracle in Eq. (11) and the prediction in Eq. (8) are computed by explicit enumeration.

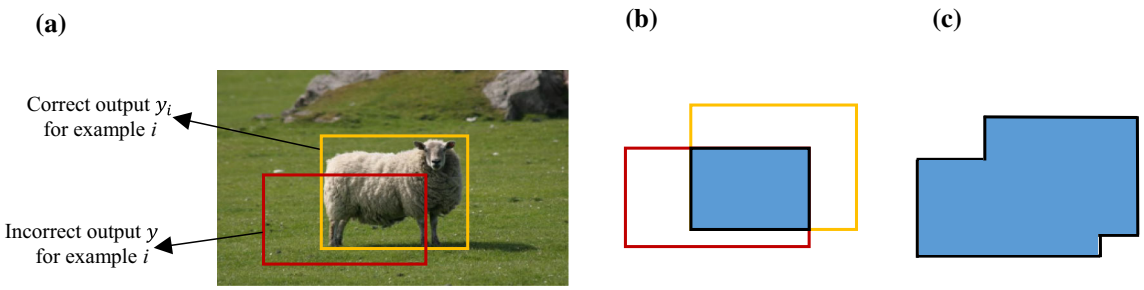

Fig. 4 **a** Correct and incorrect outputs in the image, **b** overlapped area [numerator of Eq. (14)], **c** total area of two bounding *box* [denominator of Eq. (14)]

Object detection For image analysis, object detection or localization is considered as an important task. The problem of object detection can be formulated as structured prediction problem, where the structured output is defined as a vector indicating the top, left, bottom, and right coordinates of the bounding box within the image. In our case, the input space and output space of training examples are defined as the input images and their annotations (bounding boxes) of the objects. Similar to the loss function defined in VOC challenges [36], $\Delta(y_i, y)$ is defined based on the percentage of the overlap between the ground truth (correct) bounding box of the object and other (incorrect) bounding boxes in the image as shown in Fig. 4 and given in Eq. (14).

$$\Delta(y_i, y) = 1 - \frac{\text{Area}(y_i \cap y)}{\text{Area}(y_i \cup y)} \tag{14}$$

The joint mapping function in this case is defined as the output of the hidden nodes which is $\boldsymbol{h}(\boldsymbol{x}, y)$. As mentioned in [23], the optimization process needs a maximization step for finding the most violated constraint in Eq. (11) and adding it to a working set which is used to find approximated $\boldsymbol{\beta}$. In object detection problem, this can be achieved using sliding window approach or branch-and-bound [37] approach. In this paper, we use sliding window approach where the evaluation of the objective occurs over a subset of possible bounding boxes and results in finding approximate solution. The same strategy of using sliding window approach is also used to evaluate Eq. (8) to find the highest score among possible bounding box candidates.

4.1 Benchmark data sets

In order to show the performance of the structured ELM, different data sets are used. For binary classification problems, ten UCI data sets are used and the performance is benchmarked with ELM itself. The training and testing data of the data sets of Table 1 are reshuffled at each trial of simulation.

In multi-class classification problems, another ten data sets from UCI are used. The performance is also benchmarked with ELM with single output node. Similar to binary classification data set, the training and testing data set in Table 2 are reshuffled at each trial of simulation.

Table 1 Specification of tested binary classification problems

Data sets	# Attributes	# Training data	# Testing data
Australian	6	460	230
Breast cancer	10	300	383
Colon	2000	30	32
Diabetes	8	512	256
Ionosphere	34	100	251
Leukemia	7129	38	34
Liver disorder	6	200	145
Monks problem 1	6	124	432
Monks problem 2	6	169	432
Sonar	60	100	158

For object detection problem, PASCAL VOC 2006 data set is used. The data set was proposed as a challenge to detect and classify objects from a number of visual object classes. The data set contains ten object classes in realistic scenes. The data set includes 5304 images that are evenly divided into train/validation and a test part. Since most of these images were downloaded from the Internet, it contains natural scenes that mostly include more than one object class or several instances of same/different classes. The ground truth in the data set was manually generated in the form of bounding box for each object. Example images are shown in Fig. 5. The statistics of PASCAL image sets are given in Table 3.

The performance of proposed method, structured ELM, is benched marked with structured SVM method for the given ten classes.

4.2 Features extraction

4.2.1 Binary and multi-class data sets

Binary and multi-class classification problems do not need feature extraction since the data sets are given as attributes

Table 2 Specification of tested multi-class classification problems

Data sets	# Attributes	# Ttraining data	# Testing data	# Classes
Bioinformatics	20	200	191	3
Ecoli	7	224	112	3
Glass	9	142	72	6
Iris	4	100	50	3
Segment	19	1540	770	7
Satimages	36	4435	2000	6
Traffic signals	10	300	312	6
Vehicle	18	564	282	4
Vowel	10	528	462	11
Wine	13	118	60	3

Fig. 5 Example images from the PASCAL VOC 2006 data set. Images can contain multiple object classes and multiple instances per class

of each class. So the attributes are used as features after a simple preprocessing to normalize the attributes to be between -1 and 1.

4.2.2 Object localization data sets

Following the work done in [25], the following procedures with three steps are used to get the features from images for object localization problems:

1. Bag of visual features creation

In this step, only training and validation images are used to create the bag of visual words. In each image, feature point locations are determined using salient points, regular grid, and random points. Each determined location is represented by a local descriptor, SURF [38]. Among all extracted descriptors from used images, 300,000 descriptors are sampled to build the bag of visual features using *K*-means by clustering them into 3000 visual codebook.

2. Image encoding

The feature point locations are determined in each image and represented by the SURF descriptor. Then, the matching index in the corresponding codebook for each descriptor is calculated, and it is considered as an ID of that descriptor. Finally, each image is encoded by the coordinates of the descriptors and its corresponding ID. The encoding is performed in all training, validation, and testing images.

3. Feature extraction of an object

Each object is represented as a normalized histogram with 3000 points that correspond to the visual codebook indices. The histogram is built based on the frequency of the ID's of the descriptors inside the boundary coordinates of the bounding box. Each bounding box represents either the true object's area or a non-true object's area (it can be another object of different class or another instance of the same class) inside the image. In the first case, the built histogram represents the extracted features of the true object, whereas in the second case, the histogram represents the extracted features of the non-true object inside the image. Figure 6 shows an example of bounding boxes that represent the true object and non-true object in the image.

4.3 User-specified parameters

In general, two parameters are needed to be tuned in ELM and structured ELM: the cost parameter (C) and the number of hidden nodes (L).

4.3.1 Binary classification

For each problem in binary classification, the input weight parameters $\boldsymbol{a}_i$ and the bias b_i of the hidden nodes are randomly generated from uniform distribution and assigned values between -1 and 1. The same random values settings

Table 3 Statistics of the image sets

Data sets	Train		Val		Trainval		Test	
	# Images	# Objects	# Images	# Objects	# Images	# Objects	# Images	# Objects
Bicycle	127	161	143	162	270	323	268	326
Bus	93	118	81	117	174	235	180	233
Car	271	427	282	427	553	854	544	854
Cat	192	214	194	215	386	429	388	429
Cow	102	156	104	157	206	313	197	315
Dog	189	211	176	211	365	422	370	423
Horse	129	164	118	162	247	326	254	324
Motorbike	118	138	117	137	235	275	234	274
Person	319	577	347	579	666	1156	675	1153
Sheep	119	211	132	210	251	421	238	422
Total	1277	2377	1341	2377	2618	4754	2686	4753

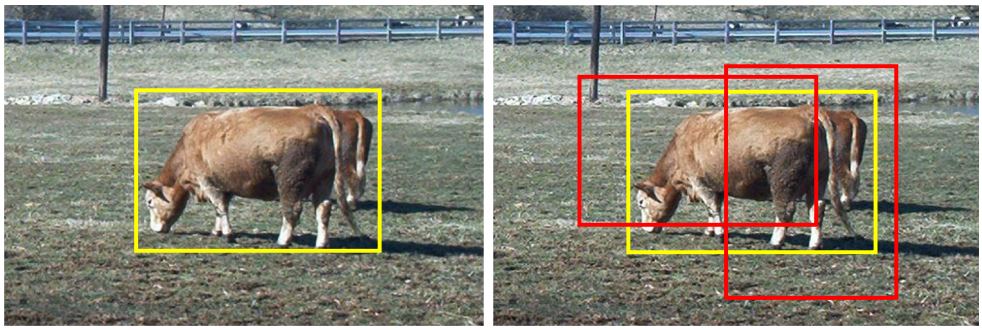

Fig. 6 Bounding *box* for object representation: true (*green*) and false (*red*) (color figure online)

are used for both ELM and structured ELM models. As it is mentioned in the literature of ELM [8, 9], the performance of ELM is insensitive to the number of hidden nodes when it is large enough (e.g., $L \geq 1000$). Therefore, L is fixed through the entire experiments and assigned to 1000 hidden nodes for both ELM and structured ELM models as in [8, 9]. Also for both models, the sigmoid function is used as activation function. In order to achieve good generalization performance and avoid over-fitting, the cost parameter C for each problem needs to be chosen appropriately. So, k-fold cross-validation method is used for both models over a wide range of values to determine the C value for each class separately. Table 4 summarizes all the setting values for ELM and structured ELM.

4.3.2 Multi-class classification

Similar to binary classification models, settings for both models are the same except for the random value. This is due to different number of input nodes. In each problem, the number of input nodes of ELM is equal to the number of attribute, where in structured ELM, the number of input nodes is equal to the number of attributes times the number of classes, i.e., $L \times K$.

4.3.3 Object localization

In object localization problems, the input weight parameters $\boldsymbol{a}_i$ and the bias b_i of the hidden nodes are assigned in a similar way like in binary and multi-class classification. To find the cost parameter C, only training data part from the data set is used for training, and then, validation portion of the data set is used for evaluation to select C. The selection of C is done separately for each class problem in the range given in Table 4. Structured ELM for object localization is benchmarked with structured SVM. In structured SVM, only the cost parameter C needed to be tuned since the linear image kernel is used. To determine C value for structured SVM, the same procedures are followed as in structured ELM with the same range of C values. For both approaches, only images that contain the object to be localized are used in the training phase.

4.4 Performance evaluation

To evaluate the performance of the proposed structured algorithm, it is benchmarked with ELM for classification and multi-class problems and with structured SVM [27] for

Table 4 ELM network parameters' values

Parameter	Binary classification problems		Multi-class classification problems		Object detection problems
	ELM	Structured ELM	ELM	Structured ELM	Structured ELM
Number of hidden nodes (L)	1000	1000	1000	1000	1000
Number of input nodes (d)	# Attributes	# Attributes	# Attributes	$d \times K$[a]	3000
$\boldsymbol{a}_i$ and b_i	Uniform distribution	Uniform distribution	Uniform distribution	Uniform distribution	Uniform distribution
Activation function	Sigmoid	Sigmoid	Sigmoid	Sigmoid	Sigmoid
Cost parameter (C)	2^{-24}–2^{25}	2^{-24}–2^{25}	2^{-24}–2^{25}	2^{-24}–2^{25}	10^{-8}–10^{4}

[a] d is the number of attributes, and K is the number of classes

Table 5 Performance comparisons of ELM and structured ELM: binary class data sets

Data sets	ELM			Structured ELM		
	C	Testing rate (%)	Testing dev. (%)	C	Testing rate (%)	Testing dev. (%)
Australian	2^{8}	75.77	2.79	2^{14}	**76.49**	2.69
Breast cancer	2^{-1}	**96.29**	0.69	2^{8}	95.94	0.77
Colon	2^{-1}	73.96	6.93	2^{2}	**74.17**	7.29
Diabetes	2^{1}	76.77	3.11	2^{10}	**76.81**	2.86
Ionosphere	2^{-2}	86.39	3.36	2^{7}	**87.07**	3.04
Leukemia	2^{-6}	83.14	9.69	2^{-1}	**86.37**	7.31
Liver disorder	2^{1}	70.44	4.04	2^{10}	**71.31**	3.54
Monks problem 1	2^{10}	75.77	3.66	2^{11}	**91.06**	2.61
Monks problem 2	2^{9}	70.95	1.85	2^{15}	**83.52**	3.16
Sonar	2^{3}	78.46	4.45	2^{7}	**78.83**	4.27

object detection problem. In binary and multi-class problems, the percentage of classification accuracy on testing data is used as a measure of the performance. The classification accuracy is defined as total correct prediction divided by the total prediction made multiplied by 100.

$$\text{Accuracy} = \frac{\#\,\text{correct prediction}}{\text{total number of predictions}} \times 100 \tag{15}$$

Thirty trials have been conducted for each problem of binary and multi-class classifications. In each trial, training and testing data sets are randomly generated and the same sets are given for ELM and structured ELM models.

In object localization problem, the performance of the proposed structured ELM is compared with structured SVM [27]. The precision measure is used as an evaluation criterion for both approaches. Precision is defined as the number of true-positive (TP) detection divided by the sum of TP and false-positive (FP) detections as given in Eq. (16).

$$\text{Precision} = \frac{\text{number}\,(\text{TP})}{\text{number}\,(\text{TP}) + \text{number}\,(\text{FP})} \tag{16}$$

A TP is accounted in the evaluation if the overlap area between the detected and the ground truth bounding boxes is >50%; otherwise, FP is accounted as shown in Eqs. (17) and (18).

$$\text{TP} = \begin{cases} 1 & \text{if overlap} \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

$$\text{FP} = \begin{cases} 1 & \text{if overlap} < 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

The whole train/validation images are used to train both structured ELM and structured SVM for the selected cost parameter. The resultant models are used to evaluate the testing sets of images. The statistical t test is also provided for the three different problems where the data analysis tool of Microsoft Excel is used to do the evaluation of the t_{Stat}.

4.5 Results and discussion

4.5.1 Binary classification problems

Table 5 shows the performance comparisons of ELM and structured ELM for binary classification data sets. The table shows the average testing accuracy rate and the standard deviation of thirty trials that are randomly generated for each class data set. Both models of ELM and

Table 6 Paired two samples for means t test in binary classification problems

	ELM	Structured ELM
Mean	78.794	82.157
Variance	62.20882667	62.91144556
Observations	10	10
Pearson correlation	0.741445357	
Hypothesized mean difference	0	
df	9	
t_{Stat}	−1.869722002	
$P(T \leq t)$ one-tail	0.047170204	
t critical one-tail	1.833112933	
$P(T \leq t)$ two-tail	0.094340408	
t critical two-tail	2.262157163	
Significance (one-tail)	Yes ($-1.869722002 < -1.833112933$)	
Significance (two-tail)	No ($-1.869722002 \not< -2.262157163$)	

Table 7 Performance comparisons of ELM and structured ELM: multi-class data sets

Data sets	ELM			Structured ELM		
	C	Testing rate (%)	Testing dev. (%)	C	Testing rate (%)	Testing dev. (%)
Bioinformatics	2^{-4}	75.13	2.87	2^{6}	**79.84**	2.71
Ecoli	2^{10}	65.51	3.45	2^{7}	**86.31**	2.81
Glass	2^{13}	53.43	4.43	2^{10}	**66.39**	5.21
Iris	2^{2}	**97.27**	1.7	2^{13}	94.13	3.48
Segmentation	2^{9}	87.09	0.84	2^{12}	**96.22**	0.7
Satimages	2^{2}	67.86	1.02	2^{13}	**91.14**	0
Traffic signals	2^{11}	28.01	2.8	3^{14}	**59.31**	2.93
Vehicle	2^{6}	72.33	2.5	2^{12}	**83.27**	2.19
Vowel	2^{9}	**91.54**	2.12	2^{12}	88.75	2.25
Wine	2^{-3}	98.22	2	2^{4}	**99.56**	1.38

structured ELM are fed with the same generated random training and testing data sets. Although structured ELM can always achieve comparable performance as ELM, it has much better performance in Monks problems one and two. The chosen cost parameter C that are used to train each model is shown in Table 5. Also, the testing rates in bold refer to higher average accuracy.

Table 6 shows the results obtained when the two-tailed paired sample t test is applied to the binary classification results shown in Table 5. The values given in the table are calculated for statistical significance $\alpha = 0.05$. As shown in Table 6, the t_{Stat} is not greater than the critical t nor it is less than the negative of the critical t. Therefore, there is no significant difference in the means between both structured ELM and ELM for binary classification problems. However, structured ELM achieves better performance in nine binary classification problems in the conducted experiments, while ELM has better performance only in breast cancer problem. This better performance can be obvious if we apply the one-tail version of the same paired t test. Table 6 shows this result where t_{Stat} is less than the negative of the critical t.

4.5.2 *Multi-class classification*

The performance comparisons for ELM and structured ELM applied to multi-class classification problems are given in Table 7. The table shows the average of testing rate accuracy and the standard deviation on 30 trials conducted for each problem using the proposed approach and ELM. Similar to binary case problems, the two models are trained and tested on the same data sets that are randomly generated. It can be seen that the performance of structured ELM is comparable to ELM and offers much better results compared to the result obtained for binary class problems. The testing rates in bold refer to higher average accuracy in Table 7. Also, the user-specified cost parameters in our simulation are given in Table 7.

The results of the statistical t test paired samples for means with $\alpha = 0.05$ are shown in Table 8. It can be seen

Table 8 *t* test paired two samples for means in multi-class classification problems

	ELM single output	Structured ELM
Mean	73.639	84.492
Variance	473.0942544	167.5625733
Observations	10	10
Pearson correlation	0.903096314	
Hypothesized mean difference	0	
df	9	
t_{Stat}	−2.985886019	
$P(T \leq t)$ one-tail	0.007651358	
t critical one-tail	1.833112933	
$P(T \leq t)$ two-tail	0.015302716	
t critical two-tail	2.262157163	
Significance (one-tail)	Yes ($-2.985886019 < -1.833112933$)	
Significance (two-tail)	Yes ($-2.985886019 < -2.262157163$)	

Table 9 Precision measure for structured ELM and structured SVM for object localization

Method	Bicycle	Bus	Car	Cat	Cow	Dog	Horse	Motorbike	Person	Sheep
ELM_1000	**0.586**	0.367	**0.404**	0.556	**0.452**	0.489	**0.343**	0.560	**0.124**	**0.357**
SVM	0.578	**0.40**	0.30	**0.571**	0.442	**0.522**	0.307	**0.607**	**0.124**	0.324

that the conditions of the significant difference are satisfied in both one-tail and two-tail tests.

4.5.3 *Object localization*

The experimental results of object localization for both structured ELM and structured SVM approaches are shown in Table 9. The results were obtained on testing data for ten different classes of PASCAL data set. A sliding window search approach was used to maximize Eq. (8) and to find the highest score of the best bounding box for the target object. In the testing phase, the object localization was only performed on testing images that contain the required object to be localized. Although many test images contain more than one instance of the object class to be detected, we consider only one object instance detection in each image for the comparison between the two approaches.

As shown in Table 9, the results of object localization with structured ELM achieve better scores compared to structured SVM in the conducted experiments. The obtained results are the precision values from the model with predefined training, validations, and testing data sets given in Table 3. In other words, the obtained results from the training model refer to one particular setup. As seen from the table, structured ELM precision score is higher than structured SVM in *five* classes out of ten classes, where structured SVM has achieved better result in *four* classes and both approaches have the same precision performance in person class.The scores in bold in Table 9 refer to higher precision values.

To get visual impression of the results obtained using structured ELM approach, Fig. 7 shows samples of the result of four different classes of object localization. Using the bounding box overlap criteria, starting from left, columns one to four show the correct object localization, whereas the last column shows the incorrect detection. The localization is considered to be true if the overlap between the detected object and the ground truth is >50% and it is drawn as green box. Red boxes are considered as a mistake in the localization where the detected box is too large or contains more than one object.

The two statistical *t* test paired sample for means are also performed, and the results are shown in Table 10 for $\alpha = 0.05$. Comparing the t_{Stat} result with *t* critical in one-tail and two-tail cases shows that there is no significance in both cases as stated in Table 10.

To test the sensitivity of structured ELM, a wide range of different values of the hidden nodes *L* and cost parameter *C* are used. These values are listed in Table 11.

In the ELM literature, the generalization performance of ELM is less sensitive to the choice of the number of hidden nodes *L* as long as it is large enough. Figure 8 shows the result of combining different values of *L* and *C* for bicycle class. Consistent with ELM theory, the performance

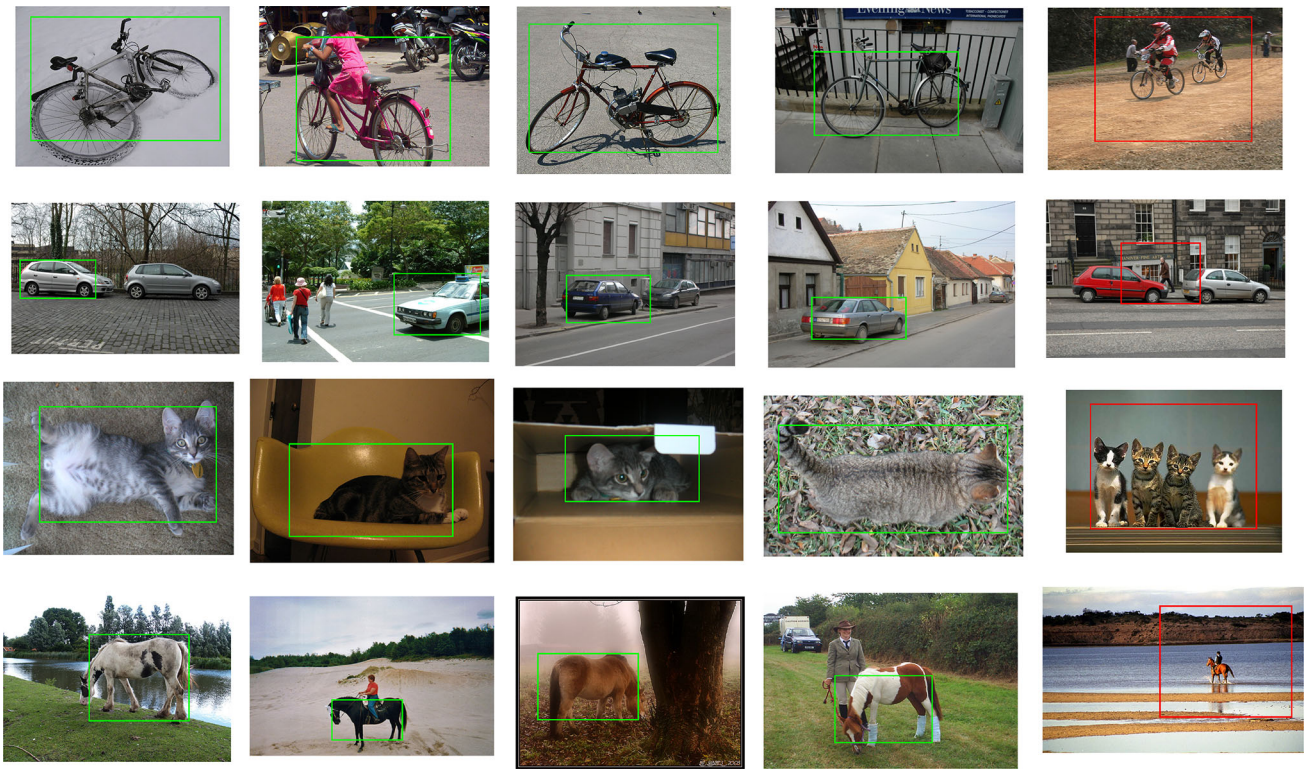

Fig. 7 Examples of the detection results for some classes using structured ELM. In the most right column, *red boxes* are counted as wrong detection by VOC evaluation routine, because the overlap with ground truth is <50% (color figure online)

Table 10 *t* test paired two samples for means in object localization problems

	Structured SVM	Structured ELM
Mean	0.4175	0.4316
Variance	0.024217833	0.019963156
Observations	10	10
Pearson correlation	0.98047197	
Hypothesized mean difference	0	
df	9	
t_{Stat}	−1.366872138	
$P(T \leq t)$ one-tail	0.10241658	
t critical one-tail	1.833112933	
$P(T \leq t)$ two-tail	0.20483316	
t critical two-tail	2.262157163	
Significance (one-tail)	No ($-1.366872138 \not< -1.833112933$)	
Significance (two-tail)	No ($-1.366872138 \not< -2.262157163$)	

Table 11 ELM network parameter's setup values

Parameter	Assigned value
Number of hidden nodes (*L*)	{250, 500,750, …, 4500}
Cost parameter (*C*)	10^{-8}–10^{4}

generally is observable to be not very sensitive with the choice of the number of hidden nodes *L* and the only parameter to be tuned in this case is *C*.

5 Conclusions

In this paper, a generalized structured ELM is proposed as extension to ELM approach to handle problems with structured outputs. Structured output is necessary for more complex problems arising in many applications such as natural language processing, bioinformatics, computer vision. Also, the proposed model treats the binary and multi-class classification problems as special

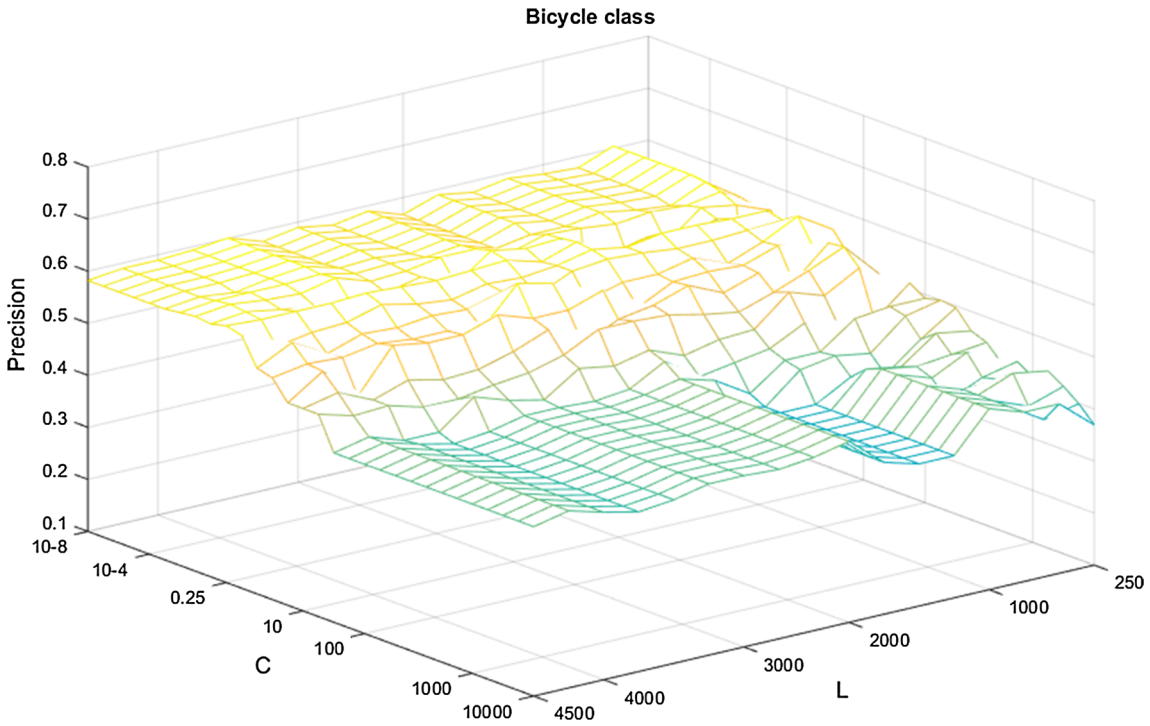

Fig. 8 Performance of structured ELM with respect to learning parameters C and L, an example on bicycle class

cases in structural learning framework. Structured ELM is formulated in standard optimization form where quadratic solution is required. "1-slack" formulation with cutting plane method is used to obtain linear time complexity in training samples number. The proposed approach performance is evaluated for different applications that have simple output such as binary and multi-class classification and more complex output such as object localization problem in still images. The simulation results show similar (for binary cases) and much better (for multi-class cases) generalization performance when compared to ELM. Also the simulation results show a comparable performance on object localization problem tested on PASCAL VOC2006 data sets when compared to structured SVM. Structured ELM wins in five classes out of ten, whereas structured SVM wins in four, and both have the same results in one class. The obtained result pointed out that structured ELM tends to be less sensitive to learning parameters, especially the number of hidden nodes, which is consistent with ELM theory. In the future, applying structured ELM to different fields of study and comparing it to the state-of-the-art methods are considered. In this paper, we only considered sigmoid as activation function, and an evaluation for different activation could also be a possible future work for different applications.

Acknowledgements This work is financially supported by Fundamental Research Grant Scheme (FRGS), VOTE 4F331 from Ministry of Higher Education, Malaysia.

References

1. Barron AR (1993) Universal approximation bounds for superpositions of a sigmoidal function. IEEE Trans Inf Theory 39:930–945
2. Leshno M, Lin VY, Pinkus A, Schocken S (1993) Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. Neural Netw 6:861–867
3. Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501
4. Huang G-B, Zhu QY, Siew C-K (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: Proceedings of the 2004 IEEE international joint conference on neural networks, pp 985–990
5. Huang G-B, Chen L, Siew C-K (2006) Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans Neural Netw 17:879–892
6. Huang G-B, Chen L (2007) Convex incremental extreme learning machine. Neurocomputing 70:3056–3062
7. Huang G-B, Chen L (2008) Enhanced random search based incremental extreme learning machine. Neurocomputing 71:3460–3468
8. Huang G-B, Ding X, Zhou H (2010) Optimization method based extreme learning machine for classification. Neurocomputing 74:155–163
9. Huang G-B, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multiclass classification. IEEE Trans Syst Man Cybern B Cybern 42:513–529

10. Liang N-Y, Huang G-B, Saratchandran P, Sundararajan N (2006) A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Trans Neural Netw 17:1411–1423
11. Huang G-B, Siew C-K (2004) Extreme learning machine: RBF network case. In: 8th control, automation, robotics and vision conference (ICARCV 2004), pp 1029–1036
12. Huang G-B, Zhu Q-Y, Mao K, Siew C-K, Saratchandran P, Sundararajan N (2006) Can threshold networks be trained directly? IEEE Trans Circuits Syst II Express Briefs 53:187–191
13. Huang G-B, Siew C-K (2006) Real-time learning capability of neural networks. IEEE Trans Neural Netw 17:863–878
14. Ding S, Xu X, Nie R (2014) Extreme learning machine and its applications. Neural Comput Appl 25:549–556
15. Frénay B, Verleysen M (2010) Using SVMs with randomised feature spaces: an extreme learning approach. In: ESANN
16. Frénay B, Verleysen M (2011) Parameter-insensitive kernel in extreme learning for non-linear support vector regression. Neurocomputing 74:2526–2531
17. Fu H, Vong C-M, Wong P-K, Yang Z (2016) Fast detection of impact location using kernel extreme learning machine. Neural Comput Appl 27:121–130
18. Rong H-J, Ong Y-S, Tan A-H, Zhu Z (2008) A fast pruned-extreme learning machine for classification problem. Neurocomputing 72:359–366
19. Sun Z-L, Choi T-M, Au K-F, Yu Y (2008) Sales forecasting using extreme learning machine with applications in fashion retailing. Decis Support Syst 46:411–419
20. Zong W, Huang G-B (2014) Learning to rank with extreme learning machine. Neural Process Lett 39:155–166
21. Schapire WWCRE, Singer Y (1998) Learning to order things. Adv Neural Inf Process Syst 10:451
22. Ailon N, Mohri M (2008) An efficient reduction of ranking to classification. In: Proceedings of the 21st Conference on Computational Learning Theory, COLT, pp 87–98
23. Joachims T, Finley T, Yu C-NJ (2009) Cutting-plane training of structural SVMs. Mach Learn 77:27–59
24. BakIr G (2007) Predicting structured data. MIT Press, Cambridge
25. Blaschko MB, Lampert CH (2008) Learning to localize objects with structured output regression. In: Computer vision—ECCV 2008. Springer, Berlin, pp 2–15
26. Hare S, Saffari A, Torr PH (2011) Struck: structured output tracking with kernels. In: IEEE international conference on computer vision (ICCV), pp 263–270
27. Tsochantaridis I, Hofmann T, Joachims T, Altun Y (2004) Support vector machine learning for interdependent and structured output spaces. In: Proceedings of the 21st international conference on machine learning, p 104
28. Schulz H, Behnke S (2014) Structured prediction for object detection in deep neural networks. In: Artificial neural networks and machine learning—ICANN 2014. Springer, Berlin, pp 395–402
29. Annema AJ, Hoen K, Wallinga H (1994) Precision requirements for single-layer feedforward neural networks. In: Proceedings of the Fourth International Conference on Microelectronics for Neural Networks and Fuzzy Systems, Turin, pp 145–151
30. Huang G-B, Babri H (1998) Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. IEEE Trans Neural Netw 9:224–229
31. Huang G-B (2003) Learning capability and storage capacity of two-hidden-layer feedforward networks. IEEE Trans Neural Netw 14:274–281
32. Rao CR, Mitra SK (1971) Generalized inverse of matrices and its applications, vol 7. Wiley, New York
33. Roller BTCGD (2004) Max-margin Markov networks. Adv Neural Inf Process Syst 16:25
34. Taskar B, Lacoste-Julien S, Jordan M (2005) Structured prediction via the extragradient method. In: NIPS
35. Tsochantaridis I, Joachims T, Hofmann T, Altun Y (2005) Large margin methods for structured and interdependent output variables. J Mach Learn Res 6:1453–1484
36. Everingham M, Zisserman A., Williams CKI, Van Gool L (2006) The pascal visual object classes (voc) challenge . Int J Comput Vis 88:303. doi: 10.1007/s11263-009-0275-4
37. Lampert CH, Blaschko MB, Hofmann T (2008) Beyond sliding windows: object localization by efficient subwindow search. In: IEEE conference on computer vision and pattern recognition (CVPR 2008), pp 1–8
38. Bay H, Tuytelaars T, Van Gool L (2006) Surf: speeded up robust features. In: Computer vision—ECCV 2006. Springer, Berlin, pp 404–417