Neural Process Lett
DOI 10.1007/s11063-014-9374-5

A Truly Online Learning Algorithm using Hybrid Fuzzy ARTMAP and Online Extreme Learning Machine for Pattern Classification

Shen Yuong Wong · Keem Siah Yap · Hwa Jen Yap · Shing Chiang Tan

Abstract This paper presents a Hybrid Fuzzy ARTMAP (FAM) and Online Extreme learning machine (OELM), hereafter denoted as FAM-OELM, which enables online learning to start from the first trained data samples without having to set up an initialization phase which requires a chunk of data samples to be ready prior to training. The idea of developing FAM-OELM is motivated by the ELM concept proposed by Huang et al., for being an efficient learning algorithm that provides better generalization performance at a much faster learning speed. However, different from the batch learning ELM and its variant called the online sequential extreme learning machine which still requires an initial offline training phase before it can turn into online training, the proposed FAM-OELM showcases a framework that enable online learning to commence right from the first data sample. Here, classification can be conducted at any time during the training phase. Such appealing feature of the proposed algorithm has strictly fulfilled the criteria of being truly sequential, while many of the existing algorithms are not. In addition, FAM-OELM automatically grows hidden neuron such that the network can accommodate new information without over fitting and compromising on the knowledge learnt earlier. The simulation results reveal the efficacy and validity of FAM-OELM when it is applied to a real world application and various benchmark problems.

S. Y. Wong · K. S. Yap (✉)
College of Graduate Studies & Department of Electronics and Communication Engineering,
Universiti Tenaga Nasional, Jalan IKRAM-UNITEN, 43009 Kajang, Selangor, Malaysia
e-mail: yapkeem@uniten.edu.my

S. Y. Wong
e-mail: joeywsy77@yahoo.com

H. J. Yap
Department of Mechanical Engineering, Faculty of Engineering, University of Malaya,
50603 Kuala Lumpur, Malaysia
e-mail: hjyap737@um.edu.my

S. C. Tan
Faculty of Information Science & Technology, Multimedia University, Jalan Ayer Keroh Lama,
75450 Bukit Beruang Melaka, Malaysia
e-mail: sctan@mmu.edu.my

Published online: 07 August 2014

Keywords Fuzzy ARTMAP (FAM) · Online sequential extreme learning machine (OSELM) · Online learning · Pattern classification

1 Introduction

Pattern classification is a subtopic of machine learning, which is defined as "the act of taking in raw data and taking an action based on the category of the data" [1]. Pattern recognition, aims at classifying data (patterns) based on either a priori knowledge that is acquired by human experts or on knowledge automatically learned from data [2]. The patterns to be classified are usually groups of measurements or observations, defining points in an appropriate multi-dimensional space [3]. Pattern recognition classifies objects into a number of classes or categories based on the patterns that objects exhibit. While two-class classification problem is well understood, multi-class classification is relatively less investigated [4]. Many pattern classification systems were developed for two-class classification problems and theoretical study of learning have focused almost entirely on learning binary functions, including the well-known support vector machines (SVM) [5] and artificial neural network algorithms such as perceptron and the error backpropagation (BP algorithm) [6]. For most of these algorithms, the extension from two-class to the multi-class pattern classification problem is non-trivial, and often leads to unexpected complexity or weaker performances [7]. Multi-class pattern recognition is a problem of building a system that accurately maps an input feature space to an output space of more than two pattern classes.

A successful classifier, as pointed out by Simpson [8], apart from yielding good performance, should possess a number of desirable properties as such: (1) to learn a required task quickly; (2) to learn new data without destroying old data (online adaptation); (3) to solve nonlinearly separable problems; (4) to provide the capability for soft and hard decisions given the degree of membership of the data within each class; (5) to provide an explanatory facility on how and why the data is classified as such; (6) to perform generalization that is independent of parameter tuning; (7) to operate without prior knowledge about the distribution of data in each class; and (8) to overcome conflicts resulted from overlapping of input space of different classes.

Among many kinds of neural networks, feedforward neural networks have gained popularity and recognition in the field of pattern classification due to their capability to approximate complex nonlinear mappings directly from the input samples. They are also prominent in providing good solutions for a large class of natural and artificial phenomena that are difficult to handle using classical parametric techniques [9]. In the literature, Multi-layer feedforward neural networks have been investigated, for example Hornik [10] attested if the activation function is continuous, bounded and nonconstant, then approximation of continuous mappings can be carried out in measure by neural networks over compact input sets. Based on the results of Hornik, Leshno [11] ameliorated and further proved that feedforward networks with a nonpolynomial activation function can approximate (in measure) continuous functions. Previously it is known that a two hidden layers can form arbitrary disjoint decision regions and a single hidden layer can form single convex decision regions. However, experimental examples suggested that a single hidden layer feedforward network (SLFN) can form decision regions that are not simply convex[12]. SLFN with any bounded continuous nonconstant activation function or any arbitrary bounded activation function which has unequal limits at infinities can form decision regions with arbitrary shapes. Since then, SLFN has proven its classification ability in a considerable number of applications [12]. Recently, a new learning algorithm for SLFN, called extreme learning machine (ELM), which is proposed by Huang [9], has drawn much attention from researchers all around. It is renowned for its learning speed can be thousands of times faster than traditional feedforward network learning algorithms like the gradient descent based method, and at the same time obtaining better generalization performance. The interest in developing and improving Extreme Learning Machine can be tracked back in few articles in [9,13–15].

The original ELM is a batch learning method which requires all training samples to be available prior to training. However, many industrial applications are incompatible with batch learning algorithms because actual industrial fields can only produce data samples gradually. Using the idea of ELM, Online Sequential Extreme Learning Machine (OSELM) [14] was developed to handle sequentially arriving data. However, OSELM has a pre-specified network structure, and consists of two training phases, i.e., an initial offline phase and a subsequent online phase. In addition, a structure-adjustable extreme learning machine (SAO-ELM) [16] was introduced. It followed the initialization phase of OSELM where an initial number of hidden neurons must be defined, followed by the online learning phase where hidden neurons can be added. Note that ELM, OSELM and SAO-ELM use randomly specified input weights and biases, and analytically determine the output weights. Random initialization may degrade performance since there may exist a set of non-optimal or unnecessary input weights and hidden biases. Several evolutionary algorithms (EA) and analytical methods have been proposed to adjust the input weights [17]. In the literature, a hybrid approach taking advantages of ELM and differential evolution (DE) was introduced to search for optimal input weights and hidden biases [17]. On one hand, Chen et al. [18] proposed a modified ELM algorithm that properly selects the input weights and biases with the Sigmoidal activation function. A recent one includes a new structure of connectionist network called the summation wavelet extreme learning machine (SW-ELM) to reduce the impact of random initialization procedure by using the well-known Nguyen Widrow [19] procedure to initialize the hidden neuron parameters [20]. Besides that, particle swarm optimization (PSO) is also presented to select and optimize the input weights and biases [21].

One of the latest developments of ELM is the pointwise ELM and pairwise ELM for learning relevance ranking problems for the first time. Here, the linear random node and linear kernel are used [22]. Besides, a new recognition algorithm that combined the discriminant tensor sub-space analysis (DTSA) and ELM is computed to improve face and micro-expression accuracy [23]. Another interesting paper [24] presents the applicability of ELM in the medical domain, i.e., magnetic resonance imaging (MRI). MRI data is the subject of active research looking for image biomarkers of neurodegenerative disorders. In this work, a procedure is introduced to discriminate patients with cocaine addiction and healthy subjects using structural MRI brain images.

In this paper, a hybrid Fuzzy ARTMAP and online extreme learning machine (FAM-OELM) is proposed. Fuzzy ARTMAP (FAM) is different from the aforementioned algorithms; it does not restrict itself to the role of properly selecting input weights and hidden biases only. FAM has the property of incremental learning, which accentuates its capability in overcoming the stability-plasticity dilemma. Besides, the network does not suffer from new data; new information can be adapted continually without a need for re-training the network with a data set that constitutes new and old data samples becomes available. The system can preserve previously learned knowledge, and, new categories can be introduced to include new information [25,26]. With these properties, FAM prepares a solid unique solution that further elevates ELM to becoming a truly online learning algorithm. The proposed FAM-OELM has following four-fold features, i.e., (1) Instead of using randomly generated input weights, the center of the hyperbox of FAM is employed; (2) The neural network starts with no hidden neuron and thus offers zero human intervention; (3) "Online learning" can be performed right from the first data sample, which means that FAM-OELM is a truly online learning algorithm [27]; (4) The number of hidden neurons dynamically increases when necessary along the training process. The proposed approach is applied to a real world application and several benchmark problems and the results show its feasibility.

The organization of the paper is as follows. Sect. 2 provides a brief introduction of FAM and OSELM, and presents the proposed FAM-OELM algorithm in detail. In Sect. 3, the performance of FAM-OELM is evaluated by comparing it with other learning algorithm. Finally, Sect. 4 offers some concluding remarks.

2 FAM, OSELM and the Proposed Hybrid FAM-OELM

This section presents the elementary introduction of FAM, and OSELM to provide the necessary background for the development of FAM-OELM.

2.1 FAM

Fuzzy ARTMAP (FAM) [18] is a supervised model from the family of the adaptive resonance theory (ART) neural network. The FAM network is an incremental learning system with dynamic structures and learning algorithm, which becomes one of the growing artificial neural network (ANN) models which succeed to overcome the stability-plasticity dilemma; i.e., FAM is stable enough to preserve information learned from previous data while being flexible enough to learn new information from new data. Besides, the network does not suffer from catastrophic forgetting [25,26]. FAM undertakes pattern classification tasks without prior knowledge on the distribution of the data samples. It is capable of self-organizing and self-stabilizing information and network configuration on presentation of input patterns [26].

2.1.1 Training Samples and Initialization

FAM network is composed of two ART modules (i.e., ART-a and ART-b). Each ART module comprises two layer of nodes; $\mathrm{F}_2^{\mathrm{a}}$ and $\mathrm{F}_2^{\mathrm{b}}$ which correspond to the inputs and target outputs, assuming the training samples presented are $\{(\mathbf{X}_1, \mathbf{Y}_1), (\mathbf{X}_2, \mathbf{Y}_2), \ldots, (\mathbf{X}_k, \mathbf{Y}_k)\}$, where $\mathbf{X}_k \in \mathbf{R}^M$ and $\mathbf{Y}_k \in \mathbf{R}^L$, are the input vector and target output vector of the k^{th} training sample, respectively. A complement coding strategy [25,26] is applied to the input and target output vectors, i.e.,

$$\begin{aligned}\mathbf{A}_k &= (\mathbf{X}_k, \mathbf{X}_k^c) = (\mathbf{X}_k, 1 - \mathbf{X}_k)\\ \mathbf{B}_k &= (\mathbf{Y}_k, \mathbf{Y}_k^c) = (\mathbf{Y}_k, 1 - \mathbf{Y}_k)\end{aligned} \tag{1}$$

where $\mathbf{A}_k \in \mathbf{R}^{2M}$ is complemented input vectors, and $\mathbf{B}_k \in \mathbf{R}^{2L}$ is complemented target output vectors.

The $\mathrm{F}_2^{\mathrm{a}}$ and $\mathrm{F}_2^{\mathrm{b}}$ are the layers where each node encodes a category of inputs and their respective target outputs, and the number of nodes can be increased when necessary during the self-organizing activity. Each node j has its own set of adaptive weights stored in the form of the vector, where the weights for FAM with N hidden neuron are $\mathbf{w}_j^a$, $\mathbf{w}_j^b$, $\boldsymbol{\mu}_j^a$, $\boldsymbol{\mu}_j^b$, which represent hyper-rectangular for ART-a and ART-b, and centers of ART-a and ART-b [26] respectively.

When a new node is added (i.e., $N \leftarrow N+1$), the weights vector are initialized to 1, while centers are set to be zero [26]. In addition, n_j is defined as the number of training sample that has been assigned to node j in $\mathrm{F}_2^{\mathrm{a}}$ layer. It is initialized as 0.

2.1.2 Competition

During learning, an input vector is presented to ART-a with its associated target output vector to ART-b. Each node j in $\mathrm{F}_2^{\mathrm{a}}$ operates winner-takes-all dynamics modeled by the choice function.

$$T_j = \frac{\left|\mathbf{A}_k \wedge \mathbf{w}_j^a\right|}{\alpha + \left|\mathbf{w}_j^a\right|} \tag{2}$$

where α is the choice parameter), and $\mathbf{w}_j^a$ is the weight vector of j node in $\mathrm{F}_2^{\mathrm{a}}$.

The winner of the competition is denoted as node J. Then, the similarity is measured between the winning prototype, $\mathbf{w}_J^a$, and $\mathbf{A}_k$ against a threshold called the vigilance parameter.

$$\frac{\left|\mathbf{A}_k \wedge \mathbf{w}_J^a\right|}{|\mathbf{A}_k|} \geq \rho_a \tag{3}$$

where $\rho_a \in [0, 1]$. If the winning node fails the vigilance test, then a new search cycle for another winning node is carried out. If no such node exists, a new node is created in $\mathrm{F}_2^{\mathrm{a}}$ to encode the $\mathbf{A}_k$.

2.1.3 Match Tracking

After the winning node J in $\mathrm{F}_2^{\mathrm{a}}$ is identified, its respective node in $\mathrm{F}_2^{\mathrm{b}}$(i.e., $\mathbf{w}_J^b$) is used to perform vigilance test in ART-b to confirm if node J is final winner.

$$\frac{\left|\mathbf{B}_k \wedge \mathbf{w}_J^b\right|}{|\mathbf{B}_k|} \geq \rho_b \tag{4}$$

$\rho_b \in [0, 1]$. If the test fails,, a match-tracking process is initiated.

$$\rho_a = \frac{\left|\mathbf{A}_k \wedge \mathbf{w}_J^a\right|}{|\mathbf{A}_k|} + 0.0001 \tag{5}$$

Insertion of a small positive value 0.0001 is to prevent reselection of node J in $\mathrm{F}_2^{\mathrm{a}}$. A new search cycle in ART-a is initiated with a new threshold level of ρ_a. If all nodes are unable to fulfill Eq. (4), a new node is added based on description in Sect. 2.1.1 training samples and initialization.

Following successful search of a winner node J, then weights and centers of winning node in $\mathrm{F}_2^{\mathrm{a}}$ and $\mathrm{F}_2^{\mathrm{b}}$ are updated as [25,27]

$$\mathbf{w}_J^a \leftarrow \mathbf{A}_k \wedge \mathbf{w}_J^a, \mathbf{w}_J^b \leftarrow \mathbf{B}_k \wedge \mathbf{w}_J^b \tag{6}$$

$$n_J \leftarrow n_J + 1 \tag{7}$$

$$\boldsymbol{\mu}_J^a \leftarrow \left(1 - \frac{1}{n_J}\right)\boldsymbol{\mu}_J^a + \frac{\mathbf{A}_k}{n_J}, \boldsymbol{\mu}_J^b \leftarrow \left(1 - \frac{1}{n_J}\right)\boldsymbol{\mu}_J^b + \frac{\mathbf{B}_k}{n_J} \tag{8}$$

2.2 OSELM

In view of ELM is a batch learning algorithm which only learn training data samples only after all training samples are available before the training, OSELM, being an extension of ELM, is proposed to deal with the emerging need for online learning in many industrial applications. OSELM is preferred over batch learning algorithm as it eliminates the step of retraining upon receiving new data [14]. OSELM can learn the training data not only one-by-one but also chunk-by-chunk with fixed or varying length. When the training has been completed, the trained data samples can be discarded.

OSELM, an online sequential learning algorithm for SLFN can learn N distinct observations with L hidden neurons and with almost any nonlinear activation function. The output function of SLFNs with respectively to $\mathbf{x}_j$ can be represented by

$$f_L(\mathbf{x}_j) = \sum\nolimits_{i=1}^{L} \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}_j) = \mathbf{t}_j \quad \text{for } j = 1, \ldots, N \tag{9}$$

where $\mathbf{a}_i$ and $\mathbf{b}_i$ are the learning parameters of the hidden neurons, β_i is the output weight, $G(\mathbf{a}_i, b_i, \mathbf{x}_j)$ denotes the output of the ith hidden node with respect to the input $\mathbf{x}_j$, and $\mathbf{t}_j$ is the target output vector. For addictive neurons with activation function g, the mathematical representation is given by

$$G(\mathbf{a}_i, b_i, \mathbf{x}_j) = g(\mathbf{a}_{i\cdot}.\mathbf{x} + b_i), \qquad a_i = \mathbf{R}^d, b_i \in R \tag{10}$$

and for RBF neurons with activation function g, it is defined as

$$G(\mathbf{a}_i, b_i, \mathbf{x}_j) = g(b_i \, \|\mathbf{x} - \mathbf{a}_i\|), \qquad a_i = \mathbf{R}^d, b_i \in R^+ \tag{11}$$

There are two phases in the training of OSELM, i.e., the initialization phase and sequential learning phase. In the initialization phase, a small chunk of training data (denoted as N_0) to train the OSELM with L (where $N_0 \geq L$) hidden neurons.

2.3 The Proposed Hybrid FAM-OELM Algorithm

As mentioned in the previous section, ELM and OSELM are quite efficient as compared to traditional methods to learn SLFN. However, due to random determination of the input weights and hidden biases, ELM-based algorithms may need a higher number of hidden neuron compared to conventional tuning-based learning algorithms in some applications which may cause ELM to respond slowly to unknown testing data [17,21]. Wang *et al.* also claimed the practice of ELM of randomly choosing the input weights and biases of the SLFN is less effective, because it is likely to produce an unexpected result that could enlarge the training error of the data samples, in which to some extent lowers the prediction accuracy [28]. Therefore, issues like parameter initialization and model complexity have to be carefully addressed for improved performance [18–20,28]. It is desired to take care of parameters initialization task and to provide a better starting point to the algorithm.

Besides that, another major bottleneck of ELM and OSELM is that they both have a pre-specified network structure. Finding an optimal size of the network in advance poses another issue to an effective learning scheme because it critically depends on the trial-and-error methods which are many times nontrivial. A fixed size network is at disadvantage because too small a network cannot learn well; too large a network structure will lead to overfitting and poor generalization performance. All the aforementioned problems can be dealt in a better manner by implementing FAM in conjunction with OSELM. FAM enjoys several desirable properties of learning such as incremental online learning as well as the property of learning stability. Owing to these inherently powerful features of FAM, ELM turns into fully online algorithm that casts off the overlaying shadow of being a partially sequential learning algorithm. FAM-OELM distinguishes itself from other ELM-based approaches such as OSELM and SAO-ELM for the network does not require an initialization phase where a set of initial training data samples have to be ready before the online learning procedures take place. In [14], OSELM made a statement that prediction can only be conducted after $L + 100$

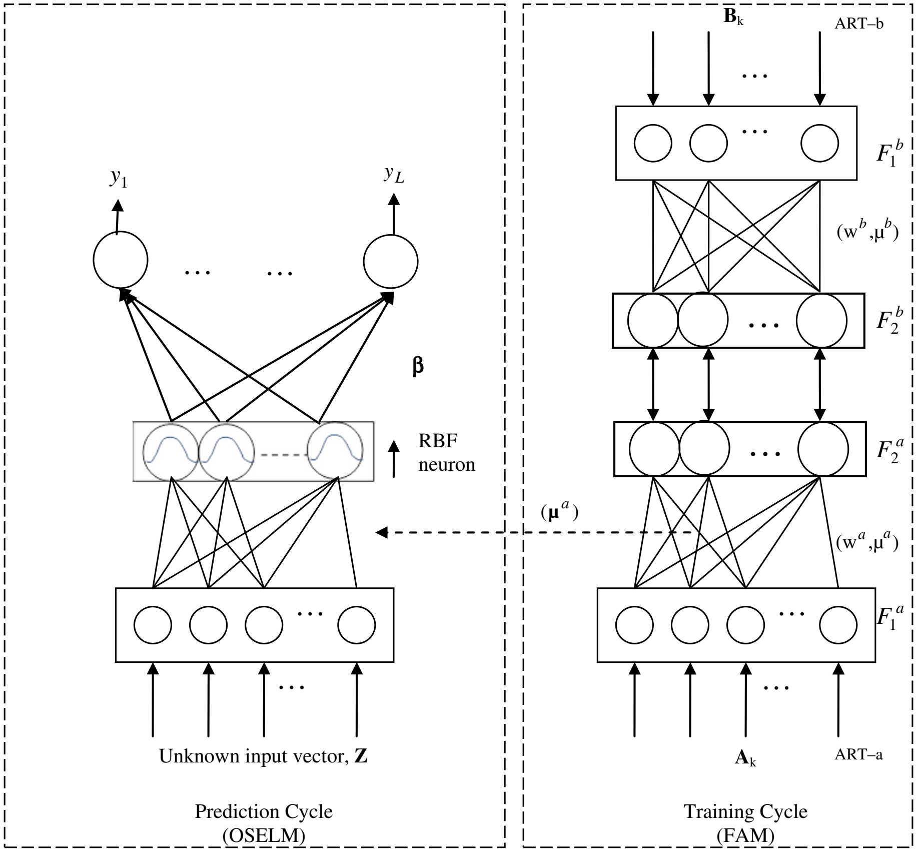


Fig. 1 The hybridization of FAM-OELM network

(L is the number of hidden neurons) for classification problems. This shows that OSELM and SAO-ELM are not fully sequential. In contrast, FAM-OELM is capable to perform online learning start from the first data samples. In addition, FAM-OELM is an intelligent computational neural network that commences with no hidden neuron and thus offers zero human intervention. The number of hidden neuron can be dynamically increased along the training process to capture necessary knowledge. FAM match-tracking mechanism attempts to avoid unnecessary creation of new neurons during the learning process and therefore controls the structural complexity of the resulting network.

The novelty of the new proposed FAM-OELM lies in its architecture which is able to grow incrementally, to accommodate input data samples sequentially, and has the ability to predict the target output for a given (unlabeled) input sample at any time during the training cycle, without any special a priori hand-crafted network structure design. And FAM-OELM also possesses the qualities of OSELM including easy implementation and able to achieve good accuracy and generalization performance.

Figure 1 shows the architecture of FAM-OELM. It should be noted that the hybridization algorithm engages the nodes of FAM be used as hidden neurons in the OELM as they are considered equivalent. As seen in the figure, $\boldsymbol{\mu}_J^a$ is the center weights of FAM, and $\boldsymbol{\beta}$ is the output weights that are computed based on the SAO-ELM [16]. The difference of the proposed network with the SAO-ELM [16] is that SAO-ELM used randomly initiated input weights where the input weights remained same throughout the training cycle, while FAM-OELM incorporates a unique self-tuning algorithm to identify a suitable center for each hidden neuron for the computation of the output matrix $\mathbf{H}$ rather. The center weights $\boldsymbol{\mu}_J^a$ from the training cycle will be passed on to the prediction cycle. Hence, it is noticeable that FAM-OELM possesses the ability to predict the target output for a given (unlabeled) input sample at any time during the training cycle.

2.3.1 The Algorithm

Consider a set of N training data samples (with an input vector and respectively target output vector), $(\mathbf{x}_j, \mathbf{t}_j) \in \mathbf{R}^n \times \mathbf{R}^m$ are used to train a FAM-OELM in one by one mode. The algorithms can be summarized as follows:

Step 1: Set training parameter of FAM-OELM, i.e., $\gamma \in R^+$ (impact factor of RBF function of OELM) and $\rho_a \in [0, 1]$ (baseline vigilance of FAM). Note that in this proposed algorithm, the impact factor b_i of RBF of Eq. (11) has been set to a user selectable parameter γ.
Step 2: For $j = 1$ to N, presents $(\mathbf{x}_j, \mathbf{t}_j)$ to FAM-OELM for one by one online learning. Go to Step 3(a) to 3(c) in the case where no new hidden neuron added by FAM. Otherwise, go to Step 4(a) to 4(b).
Step 3(a): Train $(\mathbf{x}_j, \mathbf{t}_j)$ using FAM that update the centers of neuron hidden neurons, i.e., $\boldsymbol{\mu}_k^a$ where $k = 1$ to L, and $L \leq N$.
Step 3(b): Compute hidden layer output matrix $\mathbf{H}_j$.

$$\mathbf{H}_j = [G(\boldsymbol{\mu}_1, \gamma, \mathbf{x}_j) \dots. G(\boldsymbol{\mu}_k, \gamma, \mathbf{x}_j)] \tag{12}$$

where $G(\boldsymbol{\mu}_k, \gamma, \mathbf{x}_j)$ is the output of k^{th} hidden neuron with RBF activation function respectively to the input vector $\mathbf{x}_j$ as equation below:

$$G(\boldsymbol{\mu}_k, \gamma, \mathbf{x}_j) = \exp\{-\gamma \left\| \mathbf{x}_j - \boldsymbol{\mu}_k \right\|^2\} \tag{13}$$

Step 3(c): Compute the intermediate matrix $\mathbf{K}$ and $\mathbf{P}$, and update the output weights $\boldsymbol{\beta}$. For $j = 1$, this can be done using Eqs. (14), (15) and (16).

$$\mathbf{K}_j = \mathbf{H}_j^T \mathbf{H}_j \tag{14}$$

$$\mathbf{P}_j = \mathbf{K}_j^{-1} \tag{15}$$

$$\boldsymbol{\beta}_j = \mathbf{P}_j \mathbf{H}_j^T \mathbf{t}_j \tag{16}$$

Otherwise, for $j > 1$, use equation (17), (18) and (19).

$$\mathbf{K}_j = \mathbf{K}_{j-1} + \mathbf{H}_j^T \mathbf{H}_j \tag{17}$$

$$\mathbf{P}_j = \mathbf{P}_{j-1} - \mathbf{P}_{j-1} \mathbf{H}_j^T (\mathbf{I} + \mathbf{H}_j \mathbf{P}_{j-1} \mathbf{H}_j^T)^{-1} \mathbf{H}_j \mathbf{P}_{j-1} \tag{18}$$

$$\boldsymbol{\beta}_j = \boldsymbol{\beta}_{j-1} + \mathbf{P}_j \mathbf{H}_j^T (\mathbf{t}_j - \mathbf{H}_j \boldsymbol{\beta}_{j-1}) \tag{19}$$

Step 4(a): Train $(\mathbf{x}_j, \mathbf{t}_j)$ using FAM that add a new hidden neuron with respective centers.

Table 1 Specification of benchmark datasets for experiments [14]

Datasets	# Attributes	# Class or output	# Training samples	# Testing samples
Image segmentation	18	7	1,500	810
Satellite image	36	6	4,435	2,000
DNA	180	3	2,000	1,186

Step 4(b): Compute the intermediate matrix **K** and **P**, and update the output weights $\boldsymbol{\beta}$ (including for the newly added hidden neuron) [16]. The equations are

$$\mathbf{K}_j = \begin{bmatrix} \mathbf{K}_{j-1} + \mathbf{H}_j^T \mathbf{H}_j & \mathbf{H}_j^T \mathbf{h} \\ \mathbf{h}^T \mathbf{H}_j & \mathbf{h}^T \mathbf{h} \end{bmatrix} \tag{20}$$

$$\mathbf{P}_j = \mathbf{K}_j^{-1} \tag{21}$$

$$\boldsymbol{\beta}_j = \mathbf{P}_j \begin{bmatrix} \mathbf{K}_{j-1}\boldsymbol{\beta}_{j-1} + \mathbf{H}_j^T \mathbf{t}_j \\ \mathbf{h}^T \mathbf{t}_j \end{bmatrix} \tag{22}$$

where $\mathbf{H}_j$ is the output of existing hidden neurons for $\mathbf{x}_j$ and $\mathbf{h}$ is the output of newly added hidden neuron for $\mathbf{x}_j$.

Since FAM-OELM is a fully online learning algorithm, at any time after trained with 1st data sample, it can be used for prediction purpose based on an unlabeled input vector **z**, i.e., $f(\mathbf{z}) = \sum_{k=1}^{L} \beta_k G(\boldsymbol{\mu}_k, \gamma, \mathbf{z})$.

3 Experiments and Results

In this section, the classification performance of the proposed FAM-OELM is evaluated using the three benchmark case studies [29] which include Satellite image, Image segmentation and DNA, and one of the real world applications of condition monitoring of circulating water system [30], and also on For each experiment, the dataset is divided into a training set and testing set. In the experiment, all the inputs (attributes) have been normalized into the range [0,1]. Fifty trials are conducted for each task.

A. Benchmark dataset

Three benchmark problems, as presented in [14], i.e., Image segmentation, Satellite Image, DNA, are evaluated. Table 1 details the specifications of these problems.

The satellite image problem comprises dataset generated from the Landsat multispectral scanner. There are four digital images of the same scene in four different spectral bands in one frame of the Landsat multispectral scanner imagery. The database is a tiny sub-area of a scene, consisting of 82 x 100 pixels. Each sample in the data set corresponds to a region of 3 x 3 pixels. The aim is recognize the central pixel of a region into six categories, namely red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble, and very damp grey soil given the multispectral value for each region. The training and test sets contain 4,435 samples and 2,000 samples, respectively.

The image segmentation data set consists of 2,310 regions of 3 x 3 pixels, which were randomly drawn from seven outdoor images. The objective is to classify each region into one of the seven categories, namely brick facing, sky, foliage, cement, window, path, and grass, using 19 attributes extracted from each square region.

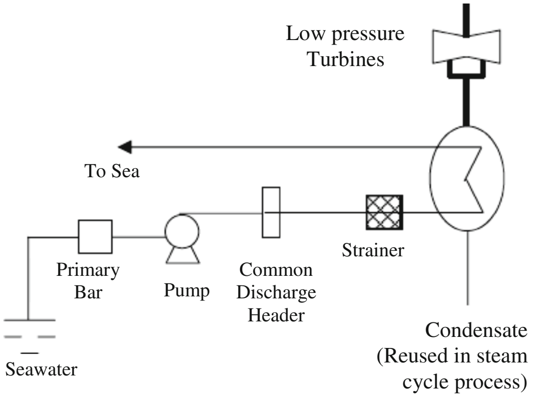
Table 2 Comparison of FAM-OELM and other learning algorithm on classification tasks

Data Sets	Algorithms	Accuracy (%)		Number of hidden
		Training	Testing	neuron
Satellite image	FAM-OELM	91.36	91.10	390
	OS-ELM(RBF) [14]	93.18	89.01	400
	GART [27]	98.04	90.53	1969
	MRAN [14]	n/a	86.36	20
	FAM [31]	n/a	89.00	704
Image segmentation	FAM-OELM	96.39	95.06	200
	OS-ELM(RBF) [14]	96.65	94.53	180
	GART [27]	99.09	96.80	634
	GAP-RBF [14]	n/a	89.93	44
	MRAN [14]	n/a	93.30	53
	FAM	98.87	93.61	220
DNA	FAM-OELM(180 features)	99.37	90.56	436
	FAM-OELM(60 features)	96.50	95.09	326
	OS-ELM (RBF) (180 features) [14]	96.12	94.37	200
	GART (180 features) [27]	100.00	88.31	1872
	GART (60 features) [27]	99.22	91.29	704
	MRAN (180 features)[14]	n/a	86.85	5
	FAM (180 features)	88.09	82.80	178
	FAM (60 features)	93.10	86.66	375

The data set of "Primate splice-junction gene sequences with associated imperfect domain theory" is known as the DNA data set [14]. Splice junctions are points on a DNA sequence at which "superfluous" DNA is removed during the process of protein creation in higher organisms. The aim of the DNA data set is to recognize the boundaries between exons (the parts of the DNA sequence retained after splicing) and introns (the parts of the DNA sequence that are spliced out) for a given sequence of DNA. This consists of three sub-tasks: recognizing exon/intron boundaries (referred to as EI sites), intron/exon boundaries (IE sites), and neither (sites). A given sequence of DNA consists of 60 elements (called "nucleotides" or "base-pairs"). Every symbolic variable representing nucleotides is coded as three binary indicator variables, thus resulting in 180 binary attributes.

Table 2 presents the classification performance of FAM-OELM and other learning algorithms. As narrated in Table 2, FAM-OELM achieves the best performance for Satellite Image dataset, with average testing accuracy of 91.10 %, a significant improvement as compared to FAM, OSELM, GART and MRAN. For Image Segmentation, the testing accuracy of FAM-OELM is better than OSELM by 0.53 % and better than FAM by 1.45 %. However, when compared to GART, GART attains better performance at the cost of hidden neuron explosion. The number of hidden neuron produced by GART is triple the FAM-OELM. As for the DNA problem, it seems that FAM-OELM does not improve the testing accuracy dramatically when compared to OSELM, however it notably outperforms GART, MRAN and FAM by 2.25, 3.71, 7.76% respectively. The result of the training set with 99.37 % accuracy implies over-training in FAM-OELM. This is likely since the data set has a huge number of input attributes, and all of them are binary values. To further examine the performance of FAM-

Fig. 2 Circulating water system



OELM, another DNA experiment, based on 60 attributes as reported in [32] is conducted. Ankerst et al. [32] pointed out that in the description of the training data, it suggested using only 60 out of 180 attributes. In fact with this means, FAM-OELM is seen obtaining the highest testing accuracy rate of 95.09 % among all other learning algorithms.

Experiments of the FAM-OELM were run in Matlab (version 7.11) environment installed in a personal computer (Operating Systems: Windows 7, CPU: Quad Core 2.9GHz, RAM: 8GB). The average training times for a sample were 5.76ms, 2.05ms, 9.28ms, 7.28ms, for Satellite Image, Image Segmentation, DNA (180 features) and DNA (60 features), respectively. As for prediction, the average time taken for predicting a sample were 0.27ms, 0.06ms, 0.93ms, 0.21ms for Satellite Image, Image Segmentation, DNA (180 features) and DNA (60 features), respectively. Both training and prediction speed were satisfactorily fast. Note that, the training and prediction are expected faster if the FAM-OELM is implemented in C/C++ codes.

B. Real world problem: Circulating water system

The system under consideration is the circulating water (CW) system of a power generation plant of Tenaga Nasional Berhad in Penang, Malaysia. The CW system plays an important role in the power generation station for providing sufficient and continuous amount of cooling water to the main turbine condensers and for condensing steam discharged from the turbine exhaust and other steam flows into the condenser. Natural sources with unlimited water supply suggest the use of oceans, bays, rivers and lakes to supply water to power plant circulating water system. Since the sea is a free and large open source of water, many power plants are located on the seacoast.

Figure 2 shows an overview of the circulating water system for a power plant [33,34]. The CW system includes all piping and equipment (such as turbine condensers and drum strainer) between seawater intake and the outfall where water is returned to the sea. The circulating water enters the plant from the sea through a primary bar screen at the pump house. The bar screen is used to prevent large sized debris, such as timber and clumps of seaweed, from entering the CW system. In the pump house, circulating water pumps draw seawater from the suction chamber to a common discharge header through a hydraulic discharge valve. From the common discharge header, the seawater flows into the CW inlet culvert through a drum strainer. This drum strainer functions as a filter to remove fine debris, such as shells and seaweed in the seawater. The circulating water flows through the culvert up to the turbine condensers, where the circulating water is used to condense steam being exhausted form the low-pressured turbine. After passing through the condensers, the circulating water exits the condenser through a pair of outlet valves into a concrete outlet culvert, before it is discharged back to the sea through an outfall.

Turbine condenser is one of the major components in the CW system. The turbine condensers use circulating water to remove rejected energy (heat) from the low-pressure steam and, at the same time, to achieve turbine backpressure (condenser vacuum) at the lowest possible yet constant level [30]. In the event that the CW system is not working at its optimum operating condition, the overall water steam cycle efficiency will be affected. Hence, Artificial Intelligence (AI) can be applied to efficiently diagnose and identify faults in the safety-related problems in the industry.

A database based on the targeted power generation of 80MW was established. The operating conditions of the CW system were categorized into four classes. Class 1 defines 'Heat transfer in the condenser is efficient and there is no significant blockage in the piping system', Class 2 denotes 'Heat transfer in the condenser is not efficient and there is no significant blockage in the piping system', Class 3 represents 'Heat transfer in the condenser is efficient and there is significant blockage in the piping system', while the final class labelled as Class 4 describes 'Heat transfer in the condenser is not efficient, and there is significant blockage in the piping system'. A set of real sensor measurements of the circulating water (CW) system was collected from the power station. This data set consists of 2,500 data samples, containing 12 attributes and is pre-divided into training and testing.

For brevity, we discussed the procedure for selecting the ρ_a (baseline vigilance parameter of FAM) and γ (impact factor of RBF). In order to determine the optimal ρ_a and γ, a cross-validation procedure is used. For all networks, i.e., ELM, OSELM and FAM-OELM, the training data samples are split into two disjoint sets, where one set is used for training and the other for validation. During the cross-validation, ρ_a of FAM is varied from 0.7 to 0.9, with interval of 0.05, whereas γ is varied from 10 to 34 with interval of 4. Table 3 summarizes the validation RMSE value and number of hidden neuron for FAM-OELM with respect to different combination of ρ_a and γ. There is no qualm to admit that one often accentuates on searching the network parameter that results in the highest validation accuracy, however the number of hidden neuron should not be overlooked. To prove the results are promising, three different cases with acceptable validation accuracy as compared to OSELM (as reported in Table 3) are selected for testing. An experiment is conducted on the unseen test data samples for fifty runs and the average results are recorded in Table 5. It is to be noted that each case delineates the number of hidden neurons deemed smaller, average and larger than the number of hidden neuron obtained from the OSELM approach, as detailed in Table 5.

To have a fair and transparent comparison, the simulations using ELM and OSELM follow closely the procedure and parameter setting as stated in [14]. The number of training data for OSELM during initialization should be about L+100 (i.e., L is the number of hidden neuron) for classification problems. ELM and OSELM requires the optimal number of hidden neuron to be determined. When conducting the cross-validation, the average accuracy rate is recorded corresponding to the number of hidden neuron tuned from 50 to 500, with an interval of 50. The number of hidden neurons is selected as the one which results in highest accuracy for the validation data set. The results of the cross-validation shown in Table 4, suggested that when the number of hidden neuron is 50, ELM and OSELM deliver the best performance. Hence L = 50 will be applied to the testing data samples.

As observed from Table 5, the smallest number of hidden neuron is accomplished by FAM-OELM Case 1 with $\rho_a = 0.7,\ \gamma = 10$, in which it records a testing accuracy (averaged over 50 runs) of 95.32 %. The performance of FAM-OELM Case 1 is similar to OSELM; however the number of hidden neurons exhibited by FAM-OELM Case 1 is only half the amount of

Table 3 The number of hidden neurons (in parenthesis) and validation accuracy of FAM-OELM

ρ_a	γ 0.70	0.75	0.80	0.85	0.90
10	(25) 95.67 %	(33) 95.92 %	(48) 96.12 %	(76) 96.43 %	(136) 97.04 %
14	(26) 95.45 %	(33) 95.74 %	(48) 96.22 %	(76) 96.47 %	(137) 97.16 %
18	(26) 95.49 %	(33) 95.74 %	(48) 96.18 %	(76) 96.50 %	(137) 97.34 %
22	(25) 95.46 %	(33) 95.61 %	(47) 96.18 %	(76) 96.62 %	(138) 97.40 %
26	(25) 95.40 %	(33) 95.61 %	(48) 96.29 %	(76) 96.57 %	(138) 97.45 %
30	(25) 93.29 %	(33) 94.51 %	(47) 95.36 %	(77) 95.88 %	(138) 97.56 %
34	(25) 93.04 %	(33) 94.47 %	(47) 95.26 %	(77) 95.89 %	(138) 96.70 %

Table 4 The results of ELM and OSELM based on validation dataset

Number of hidden neuron	ELM validation accuracy (%)	OSELM validation accuracy (%)
50	96.77	96.66
100	96.31	96.50
150	96.12	96.14
200	96.05	94.83
250	96.02	82.65
300	95.68	65.55
350	95.64	56.50
400	95.02	61.32
450	94.22	61.13
500	93.35	62.38

Table 5 The prediction accuracy of FAM-OELM, OSELM, ELM based on the test data samples

	No. of hidden neurons	Average testing accuracy (%)
FAM-OELM Case 1 ($\rho_a = 0.7,\ \gamma = 10$)	25	95.32
FAM-OELM Case 2 ($\rho_a = 0.8,\ \gamma = 26$)	48	95.83
FAM-OELM Case 3 ($\rho_a = 0.9,\ \gamma = 30$)	138	97.13
OSELM	50	95.85
ELM	50	95.80

OSELM, which results in a more compact network size. Moving on to FAM-OELM Case 2, the network produces 48 hidden neurons which can be deemed closest to the number of hidden neurons created by OSELM, and also the testing accuracy is in agreement with the result of OSELM. No need to mention, Case-3 of FAM-OELM achieves the highest testing

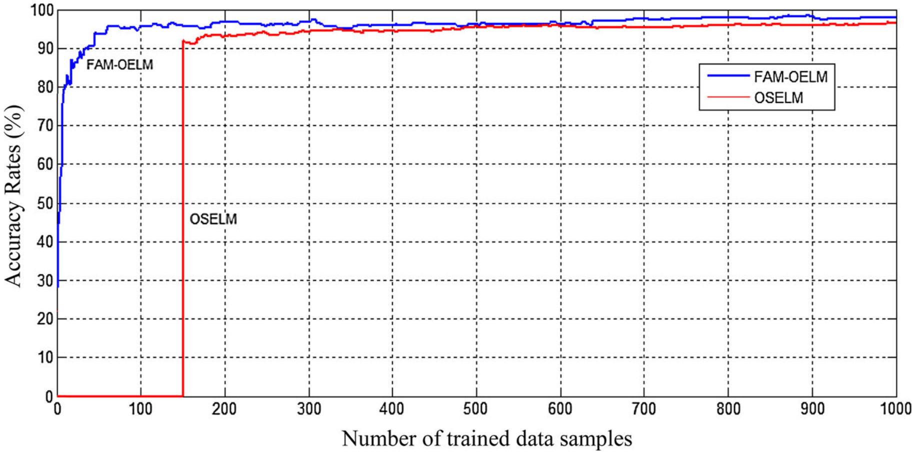


Fig. 3 The FAM-OELM prediction accuracy trending to the number of trained data samples

accuracy of 97.13 %, with 138 hidden neurons. A conclusion drawn from this table is that the FAM-OELM is capable of yielding similar testing accuracy as OSELM, at a smaller number of hidden neuron which turns out to be an advantage over the OSELM approach, but please note that the performance of FAM-OELM does not halted here. The testing accuracy rates of FAM-OELM can continue to surge higher with more hidden neurons, as elicited by FAM-OELM Case 3. This is in line with the objective of the proposed algorithm that the structure of the network is not fixed, but adjustable. On the other hand, OSELM has a fixed network structure with pre-defined number of hidden neurons which is believed to be the optimal value [14] that will deliver the best performance, therefore even if the number of hidden neuron is increased, it does not help improve the performance but causes degradation instead, as exemplified in Table 4.

To more carefully examine the relationship of online learning to the prediction outcome, the best accuracy rate of each approach, out of 50 runs for both OSELM and FAM-OELM, is selected for further analysis on the ability of how online learning varies from 1st until the last training data sample. A graph is plotted to depict the prediction trending in accordance to the increase of the trained data samples. In Fig. 3, the top blue curve corresponds to FAM-OELM with $\rho_a = 0.9,\ \gamma = 30$, and the bottom red lines is for OSELM with 50 hidden neurons.

Figure 3 clearly shows that the proposed FAM-OELM is capable of conducting efficacy prediction right after training with just one data sample, in which it has indicated the successful implementation of FAM-OELM of being a truly online learning algorithm. The accuracy rate is drastically increasing after the network is trained with more data samples. As observed from Fig. 3, when the network has been trained with 10 data samples, the testing accuracy rate for the whole test set is 79.40 %. To explain further, the prediction has improved to 90.80 % after the network is trained with 50 data samples.

On the contrary, based on [14], OSELM required initial training with $L+100$ data samples (in this case, $L = 50$) in offline mode. In other words, it is not capable to commence prediction before 150 data samples are ready (i.e., accuracy rates is 0 % before $L + 100$ data samples as evidenced in Fig. 3). One more interesting observation in this regard is that the accuracy rate of OSELM is seen at 91.90 % after the OSELM is trained with 150 data samples in the initialization phase, compared to 93.20 % captured for FAM-OELM that is also trained with the same 150 data samples. Most importantly of all, the classification strategy adopted in this

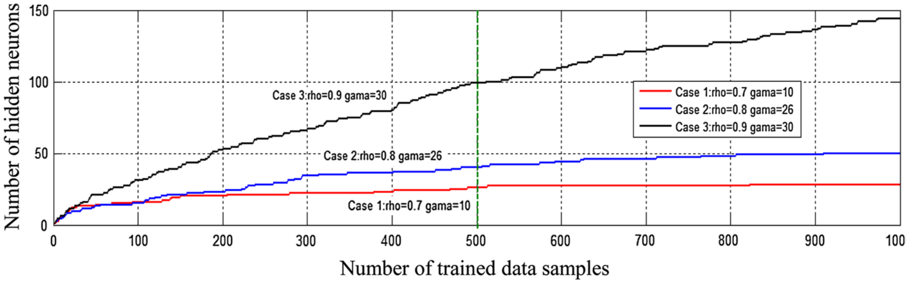


Fig. 4 Number of hidden neuron versus number of trained data samples for FAM-OELM

paper is implemented online from the beginning without the need for an initialization phase which requires a big chunk of data samples to be available prior to training.

Another interesting aspect of FAM-OELM worth analyzing is the relationship of the number of hidden neuron in response to the number of trained data samples, as portrayed in Fig. 4. The bottom red lines describe FAM-OELM Case 1with $\rho_a = 0.7$, $\gamma = 10$, the middle blue curves illustrate the FAM-OELM Case 2 with $\rho_a = 0.8$, $\gamma = 26$, and the top black lines interpret the last case of FAM-OELM with $\rho_a = 0.9$, $\gamma = 30$. There are a total of 1,000 training data samples.

As scrutinized from Fig. 4, FAM-OELM Case 1: $\rho_a = 0.7$, $\gamma = 10$ delivers 26 hidden neurons after the FAM-OELM is trained with 500 data samples. It should be noted that only two hidden neurons are later added when the network is trained with the remaining 500 data samples. The percentage increment of hidden neuron in the second half of the training cycle is only about 7.7 %.

For FAM-OELM Case 2 with rho a $\rho_a = 0.8$, $\gamma = 26$, after training with the first 500 data samples, the number of hidden neuron created is 40, an addition of ten hidden neurons is observed upon completion of 1,000 data samples. It is a gradual 25 % rise in the number of hidden neuron in the second half of the training.

Consider FAM-OELM Case 3 with $\rho_a = 0.9$, $\gamma = 30$, in the first quarter training with 250 data samples, it produces 60 hidden neurons. Moving on, 39 hidden neurons are added in the second quarter of training cycle, followed by third quarter training which appends another 26 hidden neurons. Finally when all the 1,000 training data samples are presented, 19 hidden neurons are complemented to the number of hidden neurons obtained in third quarter. The rate of hidden neurons injection is considerably declining in each quarter of the training cycle.

It makes sense to conclude that for all three cases, the increase of the number of hidden neurons in the last 500 data samples (second half of the training cycle) is much smaller as compared to the amount yield from the training of the first 500 data samples. Hence, the salient point of the proposed algorithm is that the number of hidden neurons tends to be stable or only increase at a very small rate even though more and more online training data samples are presented. Figure 5 shows the change in the increase in the number of hidden neuron along the online training process. A total of 1,000 data samples are divided into 150 intervals, displaying about six change gaps on the x-axis as shown in Fig. 5. It is obvious that the increment rate of the number of hidden neurons is getting smaller and is seen approximating to zero when the number of trained data samples increases. All in all, it is reasonable to claim that continuous online training with numerous data samples may not lead to exponential increase in the number of hidden neurons.

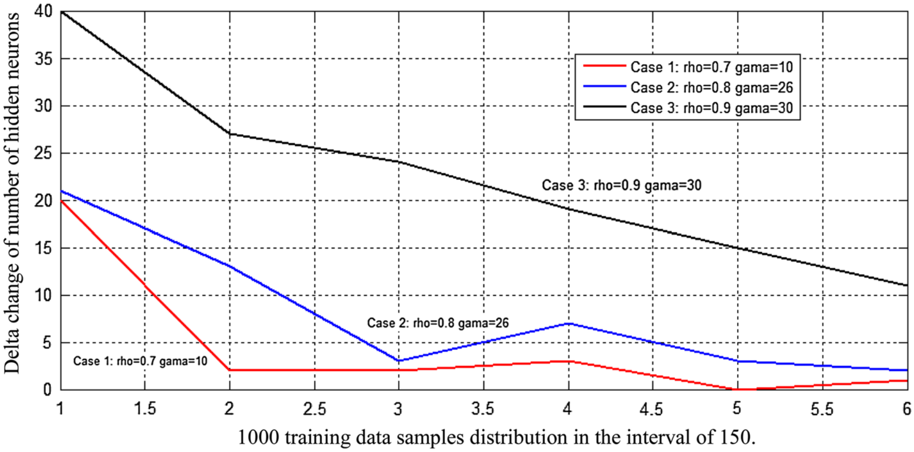


Fig. 5 Change of hidden neuron for FAM-OELM when online training

In general, the simulation results on the benchmark datasets and real world application have shown that the proposed FAM-OELM, is immensely efficient with good performance in undertaking the classification tasks.

4 Summary

In this paper, a truly online learning algorithm, denoted as FAM-OELM network, has been described. As explained, FAM-OELM does not use randomly generated input weights and bias as initial values since there may exist a set of nonoptimal or unnecessary input weights and hidden biases. FAM-OELM takes into account the parameters initialization task that provides a better starting point to the algorithm. On the other hand, ELM-based algorithms such as OSELM and SAO-ELM require an initialization phase where a chunk of training data must be ready for initial learning before online learning commences. It is worth pointing out that the proposed FAM-OELM exhibits a powerful online learning property of FAM that makes incremental learning possible right from the first data sample. Another desirable feature is that the number of hidden neuron is not predefined prior to training. Fixed size networks hinder the growth of the knowledge and may disorder the previously learnt information. The effectiveness of FAM-OELM has been demonstrated empirically using a real world application of condition monitoring, and three more benchmark datasets from the UCI machine-learning repository. The results in terms of classification accuracy and number of hidden neurons among FAM-OELM, OSELM and other popular learning algorithms have been analyzed and compared. In a nutshell, FAM-OELM which exploits the online learning capability right from the first data samples, has exhibited encouraging and promising performance in handling the classification tasks.

As for future work, it would be interesting to investigate the feasibility of transforming FAM-OELM into a rule extraction model which is capable of extracting meaningful rules with a lower degree of rule redundancy and higher interpretability within the neural network framework. In safety critical problems, strategy of formulating comprehensive rules can become very handy and indispensable in order to explain to users how a network arrives at a particular decision.

Acknowledgments This work was supported by the University of Malaya Research Collaborative Grant Scheme (PRP-UM-UNITEN), under Grant Number: CG026-2013.

References

1. Duda RO, Hart PE, Stork DG (2001) Pattern classification, 2nd edn. Wiley, New York
2. Moens M-F (2006) Information extraction: algorithms and prospects in a retrieval context, 1st edn. Springer, New York
3. Isaacs JC, Foo SY, Meyer-Baese A (2007) Novel kernels and kernel PCA for pattern recognition. In: Proceedings of the 2007 IEEE international symposium on computational intelligence in robotics and automation jacksonville, FL, U.S.A., 20–23 June, pp 438–443
4. Ou G, Murphey YL (2007) Multi-class pattern classification using neural networks. Pattern Recognit 40(1):4–18
5. Hsu C, Lin C (2002) A comparison of methods for multiclass support vector machines. IEEE Trans Neural Netw 13(2):415–425
6. Bishop CM (1995) Neural Networks for pattern recognition. Oxford University Press, Oxford
7. Anand R, Mehrotra K, Mohan CK, ranka S (1995) Efficient classification for multiclass problems using modular neural networks. IEEE Trans Neural Netw 6:117–124
8. Simpson PK (1992) Fuzzy min-max neural networks-part1: classification. IEEE Trans Neural Netw 3(5):776–786
9. Huang G-B, Zhu Q-Y, Siew C-K (2006) Extreme learning machine: theory and applications. Neurocomputing 70:489–501
10. Hornik K (1991) Approximation capabilities of multilayer feedforward networks. Neural Netw 4:251–257
11. 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
12. Huang GB, Chen Y, Babri HA (2000) Classification ability of a single hidden layer feedforward neural networks. IEEE Trans Neural Netw 11(3):799–801
13. Huang G-B, Wang DH, Lan Y (2011) Extreme learning machines: a survey. Int J Mach Learn Cybern 2(2):107–122
14. Liang NY, Huang G-B, Saratchandran P, Sundararajan N (2006) A fast and accurate online sequential learning algorithms for feedforward network. IEEE Trans Neural Netw 17(6):1411–1423
15. Huang G-B, Zhou H, Ding X, Zhang R (2012) Extreme learning machine for regression and multi-class classification. IEEE Trans Syst Man Cybern Part B Cybern 42(2):513–529
16. Li G, Liu M, Dong M (2010) A new online learning algorithm for structure-adjustable extreme learning machine. Neurocomput Comput Math Appl 60:377–389
17. Zhu Q, Qin AK, Suganthan PN, Huang GB (2005) Evolutionary extreme learning machine. Pattern Recognit 38:1759–1763
18. Chen ZX, Zhu HY, Wang YG (2013) A modified extreme learning machine with sigmoidal activation functions. Neural Comput Appl 22:541–550
19. Nguyen D, Widrow B (1990) Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In: Proceedings of the international joint conference on neural networks IJCNN, San Diego, CA, USA
20. Javed K, Gouriveau R, Zerhouni N (2014) SW-ELM: a summation wavelet extreme learning machine algorithm with a priori parameter initialization. Neurocomputing 123:299–307
21. Han F, Yao H-F, Ling Q-H (2013) An improved evolutionary extreme learning machine based on particle swarm optimization. Neurocomputing 116:87–93
22. Zong W, Huang G-B (2013) Learning to rank with extreme learning machine. Neural Process Lett 39(2):155–166
23. Wang S-J, Chen H-L, Yan W-J, Chen Y-H, Fu X (2014) Face recognition and micro-expression recognition based on discriminant tensor subspace analysis plus extreme learning machine. Neural Process Lett 39:25–43
24. Termenon M, Grana M, Barros-Loscertales A, Avila C (2013) Extreme learning machine for feature selection and classification of cocaine dependent patients on structural MRI data. Neural Process Lett 38:375–387
25. Carpenter GA, Grossberg S, Markuzon N, Reynolds JH, Rosen DB (1992) Fuzzy ARTMAP: a neural network architecture for ncremental supervised learning of analog multidimensional maps. IEEE Trans Neural Netw 3(5):698–713

26. Tan SC, Rao MVC, Lim CP (2007) A hybrid neural network classifier combining fuzzy ARTMAP and the dynamic decay adjustment algorithm. Soft Comput 12(8):765–775, Springer-Verlag
27. Yap KS, Lim CP, Abidin IZ (2008) A hybrid ART-GRNN online learning neural network with a ε-insensitive loss function. IEEE Trans Neural Netw 19(9):1641–1646
28. Wang Y, Cao F, Yuan Y (2011) A study on the effectiveness of extreme learning machine. Neurocomputing 74:2483–2490
29. Blake C, Merz C (1998) UCI repository of machine learning databases, Dept. Inf. Comput. Sci., Univ. California, Irvine, CA, [Online]. Available: http://archive.ics.uci.edu/ml/. Accessed 02 April 2013
30. Tenaga Nasional Berhad Malaysia (1999) System description and operating procedures prai power station stage 3, 14
31. Lim CP, Harrison RF (2003) Online pattern classification with multiple neural network systems: an experimental study. IEEE Trans Syst Man Cybern Part C Appl Rev 33(2):235–247
32. Ankerst M, Ester M, Kriegel HP (2000) Towards an effective cooperation of the user and the computer for classification. In: Proceeding of 6th ACM SIGKDD int. conf. on knowledge discovery & data mining (KDD-2000), pp 179–189
33. Tan SC, Lim CP (2004) Application of an adaptive neural network with symbolic rule extraction to fault detection and diagnosis an a power generation plant. IEEE Trans Energy Convers 19(2):369–377
34. Quteishat AMA, Lim CP (2008) A modified fuzzy min-max neural network with rule extraction and its application to fault detection and classification. J Appl Soft Comput 8(2):985–995