Engineering Applications of Artificial Intelligence 23 (2010) 1149–1157

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Performance enhancement of extreme learning machine for multi-category sparse data classification problems

S. Suresh [a,*], S. Saraswathi [b], N. Sundararajan [c]

[a] School of Computer Engineering, Nanyang Technological University, Singapore
[b] Laurence H. Baker Center for Bioinformatics and Biological Statistics, Iowa State University, Ames, USA
[c] School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore

Josu Maiora

Overview

The generalization performance of the ELM algorithm for sparse data classification problem depends critically on three free parameters.

1. The number of hidden neurons,
2. The input weights
3. The bias values which need to be optimally chosen.

A new, real-coded genetic algorithm approach called 'RCGA-ELM' to select them.

Two new genetic operators called 'network based operator' and 'weight based operator

We also present an alternate and less computationally intensive approach called 'sparse-ELM'.

Evaluation →A multi-class human cancer classification problem using micro-array gene expression data (which is sparse).

1. Introduction

- Single layer feedforward network (**SLFN**) with sigmoidal or radial basis activation function is found to be effective in solving a number of real world problems.

- Gradient descent algorithms for learning → Slow.

- Recently in Huang et al.(2006a,2004) and Huang (2003), it is shown that the SLFN network(with a sufficient number of H hidden neurons) with randomly chosen input weights and hidden bias can approximate any continuous function to any desirable accuracy. → Faster, good generallization: **ELM**

- We show that the **generalization performance** of the ELM algorithm for sparse data classification problem depends on the proper selection of the input weights and hidden bias values (fixed parameters) and the number of hidden neurons.

1. Introduction

- RCGA-ELM algorithm uses two different types of genetic operators namely, ‘weight based operators’ and ‘network based operators’.
- The network based operator controls the neuron growth and the weight based operator searches for the optimal weights.
- RCGA-ELM is still computationally intensive. Hence, we proposed an alternate approach called ‘sparse-ELM’ (S-ELM) which is based on K-fold validation.
- Evaluation on a real world problem of human cancer classification using the global cancer mapping (GCM) micro-array gene expression data set.

2. A brief review of extreme learning machine (ELM)

- ELM is a single hidden layer feedforward network where the input weights are chosen randomly and the output weights are calculated analytically.

In general, a multi-category classification problem can be stated in the following manner. Suppose, we have N observation samples $\{X_i,Y_i\}$, where $X_i=[x_{i1},\ldots,x_{in}]\in\Re^n$ is an n-dimensional feature of the sample i and $Y_i=[y_{i1},y_{i2},\ldots,y_{iC}]\in\Re^C$ is its coded class label. If the sample X_i is assigned to the class label c_k then kth element of Y_i is one ($y_{ik}=1$) and other elements are -1. Here, we assume that the samples belong to C distinct classes. The function which gives the necessary information on the probability of predicting the class label with the desired accuracy is called a classifier function and is defined as $Y=F(X)$. Given a known set of

2. A brief review of extreme learning machine (ELM)

- The following are the steps involved in the ELM algorithm:

  - For a given training samples(Xi,Yi), select the appropriate activation function G(·) and the number of hidden neurons H.

  - Select the input weights W and bias B randomly. Then, calculate the output weights V analytically

2.1 Issues in ELM for sparse data classification problems

- In ELM, the output weights are found analytically by minimizing the errors for randomly selected W, B and number of hidden neurons, i.e.,

$$\min_{V} \|VY_H(W,B,X) - Y\|$$

Output of the ELM network

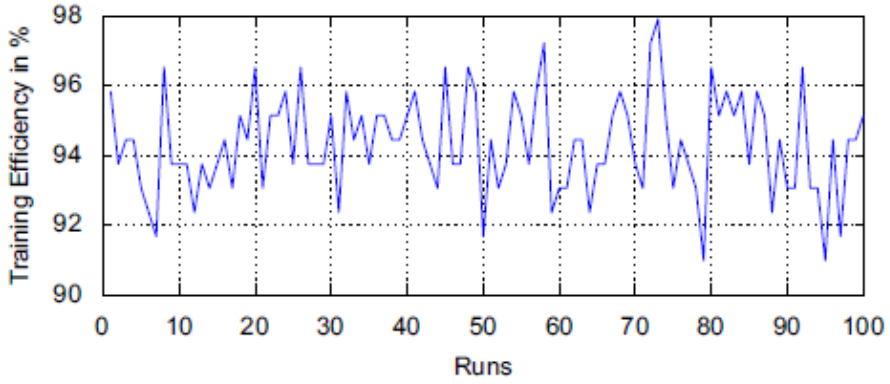
Coded Classes

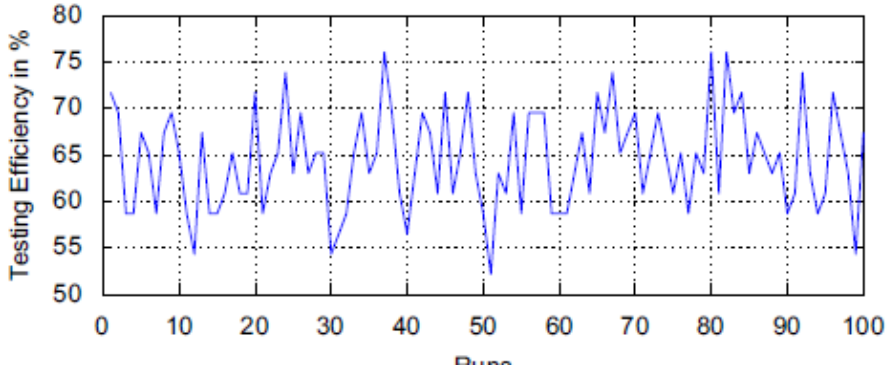
- **Given**: The input–output data and an activation function.
  Find: Optimal number of hidden neurons (H^*), corresponding input weights (W) and hidden bias values (B) such that the ELM network with analytically calculated output weights has better generalization performance.

- Particularly, the selection of the fixed parameters **affect the generalization** performance considerably in classification problems, where **fewer** number of training samples and **high imbalance** in the number of training samples per class are present.

2.1 Issues in ELM for sparse data classification problems

- We present a simulation study to analyze the behavior of ELM algorithm under **sparse** data condition using the GCM data.
- Each time the ELM algorithms is called, the fixed parameters (input weights and bias of hidden neurons) are initialized randomly using a uniform distribution.



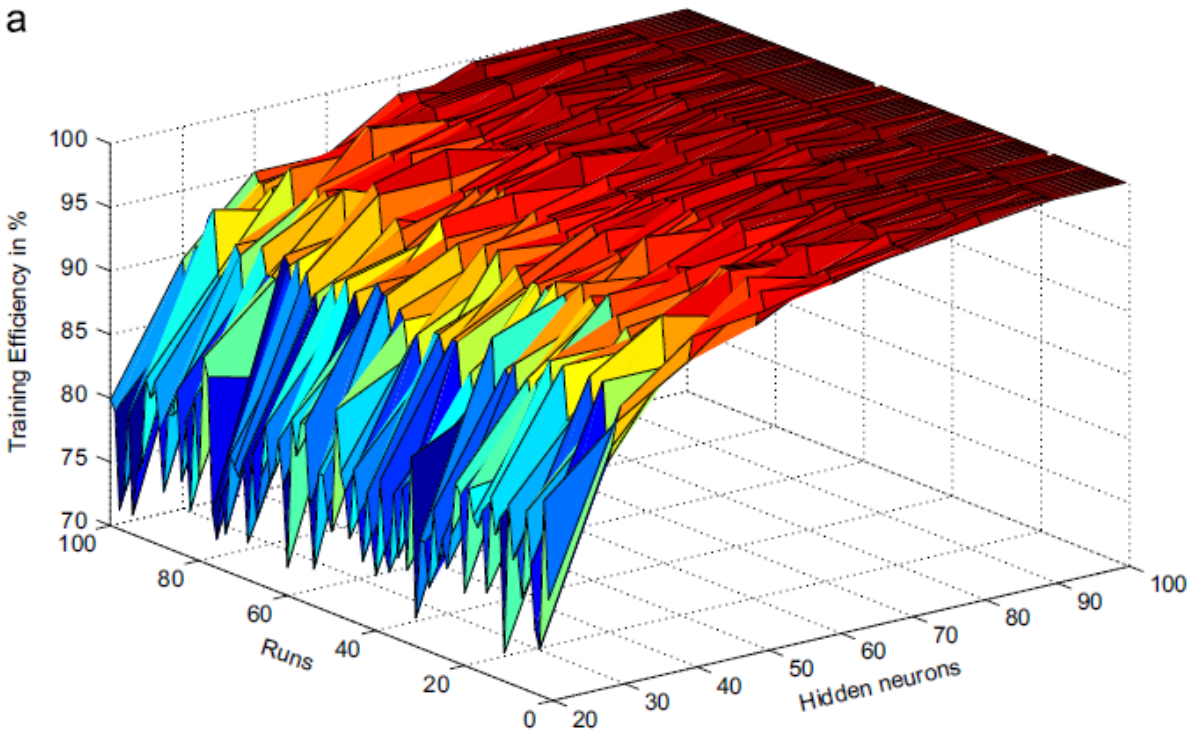


Random selection of fixed parameters results in varied performances of the ELM classifier and affects the results significantly!

40 hidden layers

2.1 Issues in ELM for sparse data classification problems

- In addition, the behavior of ELM classifier with respect to the initial parameters changes considerably with the number of hidden neurons.



It is difficult to find the best parameters (H, W and B): 2 approaches

3. A real-coded genetic algorithm approach

- The real-coded genetic algorithm (RCGA) is perhaps the most well-known of all evolution based search techniques (Michalewicz, 1994).

- We use the hybrid real coded genetic algorithm.

- Two different genetic operators are used.
  - The network based genetic operator controls the number of hidden neurons and
  - The weight based genetic operator evolves the input weight and bias values.

3. A real-coded genetic algorithm approach

- A real coded genetic algorithm for any particular optimization problem must have the following components:
  - **string representation**: the process of encoding a potential search node (solution) as a string.
  - **population initialization**:
  - **selection function**: In a genetic algorithm, new search nodes for the next generations are selected from the existing set of search nodes (population).
  - **genetic operators,**
  - **fitness function:** the fitness value is equal to the estimated cross-validation efficiency,i.e.,first, we find the output weights using analytical equation(2) with four equal datasets and evaluate the generalization efficiency of the classifier using the leave-out set.
  - **termination function:** In genetic algorithm, the evolution process continues until a termination criterion is satisfied. The most widely used termination criterion is the *maximum number of generations*

Genetic operators

- The operators are used to create search nodes based on existing search nodes in the population.

Let R and S be the two search nodes selected for crossover operations.

$$R(W,b) = \begin{bmatrix} \mathbf{W_{11}^r} & W_{12}^r & W_{13}^r & b_1^r \\ W_{21}^r & \mathbf{W_{22}^r} & W_{23}^r & b_2^r \\ W_{31}^r & W_{32}^r & \mathbf{W_{33}^r} & \mathbf{b_3^r} \\ W_{41}^r & W_{42}^r & W_{43}^r & b_4^r \end{bmatrix} \quad (10)$$

$$S(W,b) = \begin{bmatrix} \mathbf{W_{11}^s} & W_{12}^s & W_{13}^s & b_1^s \\ W_{21}^s & \mathbf{W_{22}^s} & W_{23}^s & b_2^s \\ W_{31}^s & W_{32}^s & \mathbf{W_{33}^s} & \mathbf{b_3^s} \end{bmatrix} \quad (11)$$

Genetic operators

- **Weight based operator**: In case of weight connection based crossover, L weight values are randomly selected from the parent set such that $L \in [\min(row(R), row(S))]$. This operator uses an averaging operation to generate the values of the selected connections in the children. Let $P_1 = [\mathbf{W}^r_{11}, \mathbf{W}^r_{22}, \mathbf{W}^r_{33}\, \mathbf{b}^r_3]$ be the weights selected from the parent R and $P_2 = [\mathbf{W}^r_{11}, \mathbf{W}^s_{22}, \mathbf{W}^s_{33}\, \mathbf{b}^s_3]$ be the weights selected from the parent S. The new values of

$$H_1 = P_1 + \beta(P_1 - P_2) \tag{12}$$

$$H_2 = P_2 + \beta(P_2 - P_1) \tag{13}$$

where β is a scalar value in the range of $(0 \le \beta \le 1)$. In our simulation studies, β is set to 0.2. The new children generated after the crossover operation are

$$R'(W,b) = \begin{bmatrix} \mathbf{H_1(1)} & W^r_{12} & W^r_{13} & b^r_1 \\ W^r_{21} & \mathbf{H_1(2)} & W^r_{23} & b^r_2 \\ W^r_{31} & W^r_{32} & \mathbf{H_1(3)} & \mathbf{H_1(4)} \\ W^r_{41} & W^r_{42} & W^r_{43} & b^r_4 \end{bmatrix} \tag{14}$$

$$S'(W,b) = \begin{bmatrix} \mathbf{H_2(1)} & W^s_{12} & W^s_{13} & b^s_1 \\ W^s_{21} & \mathbf{H_2(2)} & W^s_{23} & b^s_2 \\ W^s_{31} & W^s_{32} & \mathbf{H_2(3)} & \mathbf{H_2(4)} \end{bmatrix} \tag{15}$$

Genetic operators

- **Network based operator**: In case of the network based operator, we randomly select the weights of L hidden neurons from the parent set. This operator uses heuristic operation to generate the weights of the L th hidden neuron ($L \leq \min(row(R), row(S))$). Let hidden neurons selected for crossover operation be 1 (row 1 of R and S). The network weights selected for crossover operation are shown in boldface.

$$R(W,b) = \begin{bmatrix} \mathbf{W_{11}^r} & \mathbf{W_{12}^r} & \mathbf{W_{13}^r} & \mathbf{b_1^r} \\ W_{21}^r & W_{22}^r & W_{23}^r & b_2^r \\ W_{31}^r & W_{32}^r & W_{33}^r & b_3^r \\ W_{41}^r & W_{42}^r & W_{43}^r & b_4^r \end{bmatrix} \tag{16}$$

$$S(W,b) = \begin{bmatrix} \mathbf{W_{11}^s} & \mathbf{W_{12}^s} & \mathbf{W_{13}^s} & \mathbf{b_1^s} \\ W_{21}^s & W_{22}^s & W_{23}^s & b_2^s \\ W_{31}^s & W_{32}^s & W_{33}^s & b_3^s \end{bmatrix} \tag{17}$$

Genetic operators

The weights connected to the neuron 1 of parent R be P_1 (highlighted weights in R) and parent S be P_2 (highlighted weights in S). The corresponding weights values of the first hidden neuron in the children are generated as

$$H_1 = P_1 \pm \gamma w_m \frac{P_1 - P_2}{\|P_1 - P_2\|} \tag{18}$$

$$H_2 = P_2 \pm \gamma w_m \frac{P_2 - P_1}{\|P_2 - P_1\|} \tag{19}$$

where w_m is the range of the weight vectors and γ is the positive constant. In our experiment, *range* and γ are set to 2 and 0.2, respectively. The children produced after the crossover operation will be

$$R(W,b) = \begin{bmatrix} \mathbf{H_1(1)} & \mathbf{H_1(2)} & \mathbf{H_1(3)} & \mathbf{H_1(4)} \\ W^r_{21} & W^r_{22} & W^r_{23} & b^r_2 \\ W^r_{31} & W^r_{32} & W^r_{33} & b^r_3 \\ W^r_{41} & W^r_{42} & W^r_{43} & b^r_4 \end{bmatrix} \tag{20}$$

$$S(W,b) = \begin{bmatrix} \mathbf{H_2(1)} & \mathbf{H_2(2)} & \mathbf{H_2(3)} & \mathbf{H_2(4)} \\ W^s_{21} & W^s_{22} & W^s_{23} & b^s_2 \\ W^s_{31} & W^s_{32} & W^s_{33} & b^s_3 \end{bmatrix} \tag{21}$$

Genetic operators

Similar para el operador mutación: crea otro operador basado en pesos y otro basado en red.

4. A sparse-ELM algorithm

- We propose another approach using a K-fold cross-validation to select the ELM parameters quickly. The number of hidden neurons, the input weights W and B are selected using K-fold cross-validation.
- We call this algorithm 'sparse extreme learning machine' (S-ELM). The sparse-ELM is the same as the ELM algorithm with k-fold cross validation for selecting the optimal input weights and bias. The following steps are used to select the H, W and B values:

1. Randomly select a set of hidden neurons $[H_1, H_2, \ldots, H_p]$.
2. For a given H_i, use 5 $\times$ 2-fold cross-validation to select W, B.
3. Develop a classifier model using the best W and B and calculate the training and cross-validation efficiencies.
4. Repeat the steps 2 and 3 for different values of H_i; $i=1,2,\ldots,p$ and select the H_i for which the training and cross-validation efficiencies are high.
5. For the best H, W and B calculate the testing efficiency.

5. Micro-array gene expression based cancer classification problem

- Cancer detection and classification using standard clinical data are **difficult and complicated** process.
- GCM data are a collection of **micro-array gene expression** data for cancer and normal tissue specimens.
- Each tissue specimen consists of 16 063 genes and the database has 198 primary samples from 14 types of cancer.
- We use **recursive feature elimination** approach as explained in Ramaswamy et al. (2002) to select 98 of the most significant genes from the complete set of 16 063 genes

Performance evaluation

Table 1
Simulation results for cancer classification.

Type	Algorithm	H	Training efficiency		Testing efficiency		Training time in min[a]
			Mean	STD.	Mean	STD.	
SIG	ELM	45	95.12	1.32	76.50	5.31	8.20[b,d]
	S-ELM	40	94.44	1.62	88.13	4.88	30.41
	RCGA-ELM	38	94.91	1.42	89.97	4.35	270.34
RBF	ELM	50	92.30	2.25	79.43	6.23	10.20
	S-ELM	40	95.73	1.41	88.19	4.38	30.41
	RCGA-ELM	36	95.43	1.23	89.25	4.13	240.21
OVO	SVM	106[c]	96.50	1.85	73.78	5.10	29.45[b,d]

[a] MATLAB implementation.
[b] Training time includes the selection of number of hidden neurons.
[c] Number of support vectors.
[d] C++ implementation.

7. Conclusions

- One can use **sparse-ELM** approach for efficient selection of optimal parameters for function approximation and classification to achieve higher generalization performance.