Parallel Model of Online Sequential Extreme Learning Machines for Classification Problems with Large-Scale Databases

Elkin Gelvez-Almeida[1,3], Ricardo J. Barrientos[2(✉)], Karina Vilches-Ponce[2], and Marco Mora[2]

[1] Doctorado en Modelamiento Matemático Aplicado, Universidad Católica del Maule, Talca, Chile

[2] Laboratory of Technological Research in Pattern Recognition (LITRP), Facultad de Ciencias de la Ingeniería, Universidad Católica del Maule, Chile
{rbarrientos,mmora,kvilches}@ucm.cl

[3] Facultad de Ciencias Básicas y Biomédicas, Universidad Simón Bolívar, San José de Cúcuta, Colombia
elkin.gelvez@unisimon.edu.co

Abstract. Nowadays, the sizes of databases in real-world applications are around TeraByte or PetaByte. Therefore, training neural networks in reasonable times is challenging and requires high-cost computational architectures. OS-ELM is a variant of ELM, proposed for real-world applications. This algorithm allows training with new data using the previous results without reusing the previous dataset. In this work, we present a parallel model of OS-ELM for classification problems using large-scale databases. The model consists of training several OS-ELM using multi-threaded programming. The training dataset is distributed according to the number of working threads. Then, the test dataset is classified by all pre-trained OS-ELMs. Finally, the test dataset is classified using a frequency criterion. Preliminary results show that increasing the number of threads decreases the training time without significantly affecting the test accuracy of each OS-ELM.

Keywords: parallel computing · high performance computing · extreme learning machine · fingerprint classification.

1 Introduction

Extreme learning machine (ELM) is an artificial neural network (ANN) algorithm proposed for single hidden layer feedforward neural networks (SLFN). The algorithm consists of assigning random weights and biases to the hidden layer and analytically calculating the weights of the output layer using the Moore–Penrose generalized inverse matrix [3]. ELM has increased its popularity and acceptance in the scientific community due to the simplicity of the model and its generalization capacity in classification and regression problems. However, the increase in database size has been a challenge for these networks, since training times increase significantly and high-cost computational architectures are required.

Online sequential extreme learning machine (OS-ELM) is a variant of ELM proposed to address real-world problems where training data are obtained chunk-by-chunk or one-by-one [6]. This model is trained with new samples using the previous results. This methodology allows train with large-scale databases but with a high training time. OS-ELM variants that decrease the training time without affecting the accuracy were proposed in the literature. Lan et al. [5] proposed an ensemble of online sequential extreme learning machine (EOS-ELM). This algorithm constructs P OS-ELMs to form EOS-ELM, where each OS-ELM trains with new data at each incremental step. Zhai et al. [9] proposed EOS-ELM for large data set classification. Mirza et al. [7] proposed EOS-ELM for class imbalance and concept drift. Finally, Huang et al. [4] proposed a parallel EOS-ELM based on MapReduce. These previous works have inspired our proposal.

2 Online Sequential Extreme Learning Machine

In this section, we present a brief description of the preliminaries for this work starting with ELM. Let $\aleph$ be an arbitrary training set $\aleph = \{(\boldsymbol{x}_i, \boldsymbol{t}_i) | \boldsymbol{x}_i \in \mathbb{R}^n, \boldsymbol{t}_i \in \mathbb{R}^m\}$ with $i = 1, \ldots, N$, an activation function $g : \mathbb{R}^n \to \mathbb{R}^m$, and L neurons in the hidden layer with $L < N$ [3], SLFN training algorithm is given by:

$$\sum_{i=1}^{L} \boldsymbol{\beta}_i g(\boldsymbol{w}_i \cdot \boldsymbol{x}_j + b_i) = \boldsymbol{t}_j, \qquad j = 1, \ldots, N, \tag{1}$$

$\boldsymbol{w}_i$ and b_i are the i-th weights and biases of the hidden layer, respectively, $\boldsymbol{\beta}_i$ is the i-th weight of the output layer, and $\boldsymbol{w}_i \cdot \boldsymbol{x}_j$ represents the inner product of $\boldsymbol{w}_i$ and $\boldsymbol{x}_j$ [3]. Equation (3) can be written in matrix form as $\boldsymbol{H}\boldsymbol{\beta} = \boldsymbol{T}$, where $\boldsymbol{H}$ matrix is the output matrix of the hidden layer of the neural network [3]. The $\boldsymbol{\beta}_i$ weights of the output layer are calculated analytically using $\boldsymbol{\beta} = \boldsymbol{H}^{\dagger}\boldsymbol{T}$ [1], where $\boldsymbol{H}^{\dagger}$ is the Moore-Penrose generalized inverse matrix of $\boldsymbol{H}$.

We now give a brief description of OS-ELM. This algorithm is a variant of ELM for real-world applications [6]. OS-ELM has two phases: the initialization phase and the sequential learning phase. The *initial phase* uses the initial chunk of training data $\aleph_0$ with N_0 number of samples. Then, the initial weights $\boldsymbol{\beta}^0$ of the output layer are computed using the following equation:

$$\begin{cases} \boldsymbol{P}_0 = (\boldsymbol{H}_0^T \boldsymbol{H}_0 + \boldsymbol{I}/C)^{-1} \\ \boldsymbol{\beta}^0 = \boldsymbol{P}_0 \boldsymbol{H}_0^T \boldsymbol{T}_0 \end{cases}, \tag{2}$$

where H_0 is the initial output matrix of the hidden layer and C is the regularization parameter.

The *sequential learning phase* introduced the $(k+1)$-th chunks of training data. This phase makes sequential learning where $\boldsymbol{\beta}^{k+1}$ is calculated using the following equation:

$$\begin{cases} \boldsymbol{P}_{k+1} = \boldsymbol{P}_k - \boldsymbol{P}_k \boldsymbol{H}_{k+1}^T (\boldsymbol{I} + \boldsymbol{H}_{k+1} \boldsymbol{P}_k \boldsymbol{H}_{k+1}^T)^{-1} \boldsymbol{H}_{k+1} \boldsymbol{P}_k \\ \boldsymbol{\beta}^{k+1} = \boldsymbol{\beta}^k \boldsymbol{P}_{k+1} \boldsymbol{H}_{k+1}^T (\boldsymbol{T}_{k+1} - \boldsymbol{H}_{k+1} \boldsymbol{\beta}^k) \end{cases}, \tag{3}$$

The OS-ELM training is completed when the chunks of training data are finished.

3 Materials and method

In this section, we present the materials and methods used in this work. All previous experiments were carried out in a server with 2×Intel(R) Xeon(R) Gold 6238R CPUs @ 2.20GHz and 128GB of RAM, using C++ programming language and the OpenMP library for parallel programs with shared memory. This server is part of the computer cluster of the Laboratory of Technological Research in Pattern Recognition (LITRP) of the Universidad Católica del Maule, Talca, Chile.

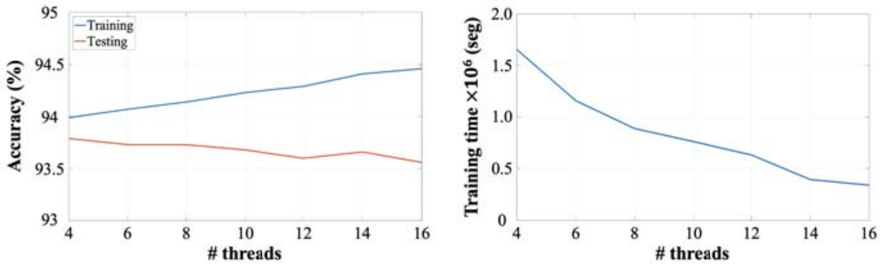
About the dataset, we used a synthetic fingerprint dataset. We employed a feature extractor to transform the fingerprints into feature descriptors. As a result of the feature extractor, vectors with 202 numerical values representing the fingerprints were generated. In addition, each descriptor (vector) has a label corresponding to the class (five fingerprint types) [8]. We balance the database for the previous experiments. Thus, the frequency of occurrence of the classes is the same (20% for class). On the other hand, we considered the neurons in the hidden layer and the regularization parameter C as hyperparameters. For the estimation, we used a data set with 60,000 samples. Experimental results show that the best accuracy is obtained with 2,000 to 5,000 neurons in the hidden layer and $C = 10$ [2]. The proposed model is presented below.

3.1 Parallel model of online sequential extreme learning machines

First, the model is related to the parallel training of a set of OS-ELM. This model consists of independent OS-ELMs that train on several threads in parallel. Then, the accuracy of each OS-ELM is averaged to compare the results using different amounts of threads. In this model, we have used a dataset with 1,000,000 samples. We used 70% of the samples for training (700,000) and 30% for testing (300,000). We divide the data set according to the number of OS-ELMs. We use a multi-thread algorithm with a one-to-one relation between threads and cores to avoid resource conflicts by the operating system [2]. Finally, we use a selection criterion based on the results of each OS-ELM. We selected the highest frequency as the selection criterion. A test dataset is introduced by each trained OS-ELM, then the highest frequency label is selected from the OS-ELMs estimations.

4 Preliminary results

In this section, we present the preliminary results of the proposed model. We used from 4 to 16 threads. The dataset was distributed according to the quantity of threads used in the experiments. The results (time and accuracy) are averaged using a simple mean to present an overview of the proposed model performance. Fig. 1a shows the training and test accuracy as the number of threads increases. The size of the datasets for each thread decreases as the threads increase since the dataset is distributed according to the threads quantity. We can see that the training accuracy increases as the threads quantity does, but the test accuracy decreases. However, the test accuracy goes from 93.79% with 4 threads to 93.56% with 16, which does not reflect a significant change.



(a) (b)

Fig. 1: Preliminary results using 5,000 neurons in the hidden layer and the regularity parameter C=10. (a) Training and testing accuracy with different numbers of threads, and (b) training time in seconds with different numbers of threads.

Regarding the training time, Fig. 1b shows the results as the threads quantity increases. Training time decreases as the threads increase. The decrease in training time is due to the threads increase, as each OS-ELM is trained with a smaller dataset size. Regarding the accuracy and training time results, we see that it significantly decreases the training time without affecting the accuracy. Finally, the accuracy is expected to increase as the threads quantity increase using the frequency criterion.

5 Conclusions

This paper proposed a parallel model of online sequential extreme learning machines for classification problems with large-scale databases. This model distributed the dataset in various OS-ELMs trained in parallel. Then, the testing dataset is classified using a frequency criterion according to the results of each OS-ELM. The preliminary results show that the training time decreases when the threads number increases. Testing accuracy also is not affected when the threads quantity increases. The preliminary results suggest that it is appropiate to increase the threads quantity to decrease the training time, since it does not affect the accuracy of the test. However, it requires a computational architecture with a suitable core quantity. Finally, we expected that the accuracy to increase when applying the frequency criterion.

5.1 Future work

In future work, we plan to increase the threads number to identify the moment that decreases the accuracy. In addition, we will analyze the results using this unbalanced data set, simulating a population in the real world. We will also use other databases and ELM variants to test our model in different applications. In addition, we will propose a parallel mathematical model to compute the Moore-Penrose generalized inverse using multiple threads. This model aims to decrease the training time of ELM and their variants since it is the slowest procedure. Finally, computing the Moore-Penrose generalized inverse with a parallel model will allow us to design an OS-ELM model based on a parallel processing strategy as a pipeline.

Acknowledgement

This work was funded by the National Agency for Research and Development (ANID) / Scholarship Program / NATIONAL DOCTORAL SCHOLARSHIPS / 2020 - 21201000. The authors of the paper also thank Research Project FONDECYT REGULAR 2020 Nº 1200810 "Very Large Fingerprint Classification Based on a Fast and Distributed Extreme Learning Machine," National Agency for Research and Development (ANID), Government of Chile.

References

1. Gelvez-Almeida, E., Baldera-Moreno, Y., Huérfano, Y., Vera, M., Mora, M., Barrientos, R.: Parallel methods for linear systems solution in extreme learning machines: an overview. In: Journal of Physics: Conference Series. vol. 1702, p. 012017. IOP Publishing (2020)
2. Gelvez-Almeida, E., Barrientos, R.J., Vilches-Ponce, K., Mora, M.: Parallel training of a set of online sequential extreme learning machines. In: 2022 41st International Conference of the Chilean Computer Science Society (SCCC). pp. 1–4. IEEE, Santiago (2022)
3. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: A new learning scheme of feedforward neural networks. In: 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No. 04CH37541). vol. 2, pp. 985–990. IEEE, Budapest (2004)
4. Huang, S., Wang, B., Qiu, J., Yao, J., Wang, G., Yu, G.: Parallel ensemble of online sequential extreme learning machine based on MapReduce. Neurocomputing **174**, 352–367 (2016)
5. Lan, Y., Soh, Y.C., Huang, G.B.: Ensemble of online sequential extreme learning machine. Neurocomputing **72**, 3391–3395 (2009)
6. Liang, N.Y., Huang, G.B., Saratchandran, P., Sundararajan, N.: A fast and accurate online sequential learning algorithm for feedforward networks. IEEE Transactions on Neural Networks **17**, 1411–1423 (2006)
7. Mirza, B., Lin, Z., Liu, N.: Ensemble of subset online sequential extreme learning machine for class imbalance and concept drift. Neurocomputing **149**, 316–329 (2015)
8. Zabala-Blanco, D., Mora, M., Barrientos, R.J., Hernández-García, R., Naranjo-Torres, J.: Fingerprint classification through standard and weighted extreme learning machines. Applied Sciences **10**, 4125 (2020)
9. Zhai, J., Wang, J., Wang, X.: Ensemble online sequential extreme learning machine for large data set classification. In: 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 2250–2255. IEEE, San Diego (2014)