A Secure and Practical Mechanism of Outsourcing Extreme Learning Machine in Cloud Computing

Jiarun Lin, Jianping Yin, Zhiping Cai, Qiang Liu, Kuan Li

***Abstract*—The enlarging volume and increasingly complex structure of data involved in applications makes Extreme Learning Machine(ELM) over large-scale data a challenging task. The paper presents a secure and practical mechanism for outsourcing ELM in cloud computing, named Partitioned ELM, to reduce the training time while assuring the confidentiality of the input and the output. The cloud server is mainly responsible for calculating the Moore-Penrose generalized inverse which is the heaviest operation computationally. The inverse also serves as the correctness and soundness proof in result verification. We analyze the confidentiality theoretically and the experimental results demonstrate that the proposed mechanisms can effectively release customer from heavy computations. The customer can further speedup the ELM by outsourcing multiple ELM problems simultaneously in cloud computing.**

Keywords*—*Extreme Learning Machine, Cloud Computing, Computation Outsourcing, Partitioned ELM

I. Introduction

Extreme Learning Machine(ELM) [1], [2], [3], [4] , is a newly proposed learning algorithm for generalized Single-hidden Layer Feedforward Neural networks(SLFNs). The enlarging volume and increasingly complex structure of data involved in applications makes Extreme Learning Machine(ELM) over large scale data a challenging task. To address the challenge, researchers have proposed many enhanced ELM variants[5], [6]. However, users who own the large scale data may not have abundant computing resources or distributed computing frameworks in hand. Instead, we can outsource the expensive computation of ELM in cloud computing to utilize the literally unlimited resources in a pay-per-use manner at relatively low prices.

To the best of our knowledge, we are the first to outsource Extreme Learning Machine algorithms in cloud computing while assuring the confidentially of the input training samples and the output results. ELM problems, in which the parameters of hidden nodes in ELM are assigned randomly and the desired output weights can be determined analytically, are suitable for being outsourced in cloud computing, i.e., the input training samples can be well protected without additional encryption. The confidentiality can be instinctively assured.

This paper proposes a secure and practical outsourcing mechanism, named Partitioned ELM to address the challenge of performing ELM over large scale data. It explicitly decomposes ELM algorithm into public part and private part. The public part is executed in the cloud server, mainly responsible for the calculation of Moore-Penrose generalized inverse, which is the most time-consuming calculations of ELM. The private part consists of generations of random parameters and some light matrix operations. The generations of random parameters are associated with the confidentiality of input data. With the random parameters, the customer calculates the output matrix of the hidden layer from which cloud server cannot mine out sensitive information. The cloud computing calculates the Moore-Penrose generalized inverse of the matrix and sent it back to the customer. Then the customer transforms it to the desired result while preserving the security and privacy of the input training samples and the desired output.

Extensive experiments are conducted to evaluate the performance of the proposed mechanism. And the experimental and analytical results show that the proposed mechanisms can save considerable training time of the ELM. When the size of the ELM problem increases, the speedups achieved by the proposed mechanism are also getting larger.

Jiarun Lin, Jianping Yin, Zhiping Cai, Qiang Liu and Kuan Li are with the School of Computer, National University of Defense Technology, Changsha, P.R. China 410073. Email: nudtjrlin@gmail.com

II. Brief Review of ELM

ELM is first proposed by Huang et al [1], [2], [3], [4]. It has been proved that adjusting the input weights $\mathbf{w}$ and the biases $\mathbf{b}$ iteratively is not necessary. Instead, they can be randomly assigned if the activation functions in the hidden layer are infinitely differentiable. The output weights β can be determined analytically through simple generalized inverse operation of the hidden layer output matrices.

The N arbitrary distinct samples are modeled by matrixes (X, T), whose dimensions are $N \times n$ and $N \times m$ respectively, where n denotes the number of input attributes and m denotes the number of target labels. The weight parameters ($\mathbf{w}$,$\mathbf{b}$) are assigned randomly and the smallest norm least-squares solution of the output weights $\beta = H^{\dagger}T$ where H is the output matrix of the hidden layer and $H^{\dagger}$ is the Moore-Penrose generalized inverse of matrix H[7].

III. Outsourcing ELM in Cloud Computing

A. Threat Model of Cloud Computing

In order to reduce the time used for training or executing ELM on large scale data, it is intuitional to outsource the bottle-neck computation to cloud computing with abundant resource. However, outsourcing ELM in cloud is also relinquishing user's direct control of their data and application, in which sensitive information might be contained.

The threats of confidentiality in the outsourcing of ELM mainly stem from the cloud computing. The cloud computing may behave in "honest-but-curious" model, which is also called semi-honest model that was assumed by many previous

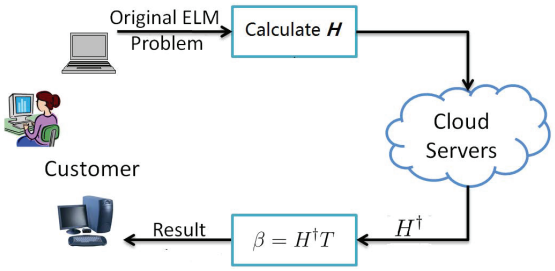

Fig. 1: Architecture of Outsourcing ELM in Cloud Computing

researches[8]. The cloud server may be persistently interested in analyzing the data to mine more information for various purposes, either because it intends to do so or because it is compromised. In this paper, we assume that the cloud servers may behave unfaithfully, beyond semi-honest model, i.e., it may cheat to the customer to save power or reduce executing time while hoping not to be caught at the same time. To enable secure and practical outsourcing ELM in cloud computing, the proposed mechanisms should be ingeniously designed so as to ensure the confidentiality of ELM problems while guaranteeing the correctness and soundness. We firstly assume that the cloud server performs the computation honestly and discuss the verification of correctness and soundness later.

B. Architecture of Partitioned ELM

There are two different entities involved in the outsourcing ELM in cloud computing: the cloud customers and the cloud servers in the cloud computing. The former entity has several computationally expensive large scale ELM problems to outsource in cloud computing, and the latter one has literally unlimited resources and provides utility computing services. The architecture for outsourcing ELM in cloud computing is illustrated in Figure 1.

To focus on the outsourcing ELM in cloud computing, we assume that the communication channels between the cloud server and the customers are reliably authenticated and encrypted, which can be achieved in practice with little overhead. So the authentication processes are omitted in this paper.

It explicitly decomposes ELM algorithm into public part and private part, as the name Partitioned ELM indicates. The public part is executed in the cloud server, mainly responsible for the calculation of Moore-Penrose generalized inverse, which is the most time-consuming calculations of ELM. The private part consists of generations of random parameters and some light matrix operations. The generations of random parameters are associated with the confidentiality of input data. With the random parameters and randomly chosen activation functions, the customer calculates the output matrix of the hidden layer from which cloud server cannot mine out sensitive information. Besides, the customer multiply the inverse with the target matrix to calculate the desired output weights.

C. Encryption of Training Samples

ELM determines the parameters for the neural networks at different steps. The ELM is instinctively suitable for outsourcing in cloud computing while assuring the confidentiality of the training samples and the desired parameters of neural networks.

In the private part, more specifically, the parameters $(\mathbf{w},\mathbf{b})$ are assigned randomly which are a part of the desired outputs of the training neural networks. These parameters must be assigned by the cloud customer but not the cloud server.

The universe of infinitely differentiable activation functions is infinite. Without any knowledge of the activation function or the parameters, the cloud server can not obtain any knowledge about the exact X or $(\mathbf{w}, \mathbf{b})$ from H. So the encryption of the X is embedded in the ELM and the confidentiality of the input training samples and training neural network's parameters $(\mathbf{w}, \mathbf{b})$ is achieved by the randomly generated parameters and randomly chosen activation functions.

For convenience, we denote $H = \mathbf{g}(H_0)$. Noting that even with the knowledge of the infinitely differentiable activation functions associated with the hidden nodes, the cloud server cannot exactly figure out $X, \mathbf{w}$, or $\mathbf{b}$ from the mediate matrix H_0. Therefore, we also can outsource the computation of the activation functions in cloud computing. The communication overhead between the customer and the cloud server can be further reduced using pipeline parallelization, i.e., the cloud server calculates the activation functions and receives H_0 in a pipeline manner.

D. Calculation of Output Weights

The cloud server receives the mediate matrix H_0 and then calculate the output matrix of the hidden layer. Thereafter, it calculate the Moore-Penrose generalized inverse, whose execution time dominates the training time of the original ELM problem and sends the Moore-Penrose generalized inverse back to the customer. Finally, the customer calculates the output weights β by multiply the inverse $H^{\dagger}$ and the target output T of the training samples locally.

In the whole process, the parameters $((\mathbf{w}, \mathbf{b}, \beta)$ of the training neural networks are kept out of the reach of the cloud server. The cloud server cannot mine out special information about the original ELM problems and the trained neural networks, such as the training samples (X, T) and the desired parameters.

In this paper, we only focus on outsourcing the basic ELM algorithm in cloud computing. It is worth noting that, the proposed mechanisms are not limited to a specific type of ELM and can be employed for a large variety of ELMs. Applying our outsourcing mechanisms to various ELM variants is one of our future works.

E. Result Verifications

Till now, we have been assuming that the server is honestly performing the computation, while being interested learning information. However, the cloud server might behave unfaithfully. Therefore, the customer must be enabled to verify the correctness and soundness of the results.

TABLE I: Performance over Part of the CIFAR-10 Dataset

M	$t_{original}$(s)	$t_{outsource}$(s)	$t_{customer}$(s)	t_{cloud}(s)	λ
500	12.65	6.19	2.70	3.48	4.69
1000	53.94	17.07	5.07	12.00	10.64
1500	114.29	33.62	7.46	26.16	15.32
2000	347.02	57.840	10.10	47.74	34.36
2500	485.30	89.78	12.58	77.20	38.58
3000	1055.95	135.74	14.79	120.95	71.40
3500	1513.80	191.40	17.29	174.11	87.55

In our mechanism, the returned inverse itself from the cloud server can also serve as the verification proof. From the definition of Moore-Penrose generalized equations we can verify whether the returned matrix is the desired inverse. Therefore, the correctness and soundness of the results can be verified while incurring few computation overhead or extra communication overhead.

IV. Performance Evaluation

We use $t_{original}$ to denote the training time of the original ELM and $t_{outsource}$ to denote that of the proposed mechanism. In the Partitioned ELM, the time cost at the customer side and at the cloud server side are denoted as $t_{customer}$ and t_{cloud}, respectively. Then we define the *asymmetric speedup* of the proposed mechanism as $\lambda = \frac{t_{original}}{t_{customer}}$. The physical meaning of λ is the savings of the computing resources for the customer. The asymmetric speedup is independent on how resourceful the cloud server is and directly related with the size of ELM problems. Through outsourcing the calculation of Moore-Penrose generalized inverse in cloud computing with resourceful computing power(e.g., CPU, memory), the time cost at the customer side would be reduced dramatically.

In the series of experiments, the customer computations in our experiments are conducted on a workstation with an Intel Xeon Quad Processor running at 3.60GHz with 2GB RAM and 1GB Linux swap space while the cloud server computations on a workstation with an Intel Core Duo Procession running at 2.50GHz with 4GB RAM and huge enough Windows Virtual Memory. Through outsourcing the bottle-neck computation of ELM from a workstation with lower resource to another workstation with more computing power, we can evaluate the training speed of the proposed mechanisms without a real cloud environment. Our proposed mechanism focus on improving the training speed through outsourcing while the training accuracy and testing accuracy are not affected.

We test the Partitioned ELM over a large scale dataset named CIFAR-10 [9] which consists of 50000 32×32 training color images and 10000 testing images in 10 classes. There are 5000 training images and 1000 testing images per class. To reduce the number of attributes, we transform the color images into gray images. We conduct 5 trials for each M, and randomly choose 2 classes from the 10 classes as the training samples and testing samples for each trial.

The results are listed in Table I. With the increase of M, memory is becoming the dominant computing resource when solving the ELM problem. And the asymmetric speedup is also increasing. It means that the larger the problems's size, the larger speedups the proposed mechanism can achieve. When M is substantially large, the original ELM will terminate due to the memory limit.

The training accuracy is also inclining steadily from 83% to 95% along the number of hidden node while the testing accuracy changes between 80% and 84%. To find specific M for the ELM problem with best testing accuracy, one may wish to test multiple experiments under different values of M. Then the resourceful computing power of the cloud computing can be more fully utilized in the way that the cloud server tests multiple ELM problems with different M simultaneously to reduce the overall training time.

V. Conclusion

The Partitioned ELM explicitly decomposes ELM problems into two parts to address the challenge of performing ELM over large scale dataset. The customer assigned the input weights and the bias of the training neural networks locally. The cloud server is mainly responsible for calculating the Moore-Penrose generalized inverse, which also serves as the verification proof. Thereafter, it is sent back to the customer who multiplies it with target matrix locally to get the output weights. The experimental results show that Partitioned ELM can release the customer from heavy burden of expensive computations, achieving higher asymmetric speedup for the larger ELM problems.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Project no. 60970034, 61170287, 61232016, 61070198).

We thank Dr. Guang-Bin Huang and the reviewers for their constructive and insightful comments of this paper.

References

[1] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, pp. 489–501, 2006.

[2] G.-B. Huang, L. Chen, and C.-K. Siew, "Universal approximation using incremental constructive feedforward networks with random hidden nodes," *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.

[3] G.-B. Huang and L. Chen, "Convex incremental extreme learning machine," *Neurocomputing*, vol. 70, pp. 3056–3062, 2007.

[4] G.-B. Huang and L. Chen, "Enhanced random search based incremental extreme learning machine," *Neurocomputing*, vol. 71, pp. 3460–3468, 2008.

[5] Q. He, T. Shang, F. Zhuang, and Z. Shi, "Parallel extreme learning machine for regression based on mapreduce," *Neurocomputing*, 2012.

[6] M. van Heeswijk, Y. Miche, E. Oja, and A. Lendasse, "Gpu-accelerated and parallelized elm ensembles for large-scale regression," *Neurocomputing*, vol. 74, no. 16, pp. 2430–2437, 2011.

[7] D. Serre, *Matrices: Theory and applications*, vol. 216. Springer, 2010.

[8] C. Wang, K. Ren, and J. Wang, "Secure and practical outsourcing of linear programming in cloud computing," in *INFOCOM, 2011 Proceedings IEEE*, pp. 820–828, IEEE, 2011.

[9] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Master's thesis, Department of Computer Science, University of Toronto*, 2009.