ISSN 2250-3153

Privacy Preserving Back Propagation Algorithm for Distributed Neural Network Learning

Mr. S. S. Sayyad, Dr. P.J. Kulkarni

Department of Computer Science & Engineering
Walchand College of Engineering, Sangli, India

Abstract- We address to the problem of Privacy Preserving Back Propagation Algorithm for a Vertically Partitioned Dataset. To enhance cooperation's in learning, it is important to address the privacy concern of each data holder by extending the privacy preservation notion to original learning algorithms. In this paper, we focus on preserving the privacy in an important learning model, multilayer neural networks. We present a privacy-preserving multiparty distributed algorithm of back propagation which allows a neural network to be trained without requiring either party to reveal her data to the others. We provide complete correctness and security analysis of our algorithms. The effectiveness of our algorithms is verified by experiments on various real world data sets.

Index Terms- back propagation, privacy preserving, multi-party computation, neural network, ElGamal Scheme.

I. INTRODUCTION

With the development of distributed computing environment, many learning problems now have distributed input data. In such distributed scenarios, privacy concerns often become a big issue. In recent years, Secure Multiparty Computation (SMC) [1] and privacy preservation have attracted much attention in incorporating security into data mining and machine learning algorithms [2], [3], [4], [5], [6], [7]. A natural question is why the researchers would want to build a learning model (e.g., neural networks) without first collecting all the training data on one computer. If there is a learner trusted by all the data holders, then the trusted learner can collect data first and build a learning model. However, in many real-world cases, it is rather difficult to find such a trusted learner, since some data holders will always have concerns like "What will you do to my data?" and "Will you discover private information beyond the scope of research?" On the other hand, given the distributed and networked computing environments nowadays, collaborations will greatly benefit the scientific advances.

In this paper, we focus on one of the most popular techniques in machine learning, multilayer neural networks [8], [9], in which the privacy preservation problem is far from being practically solved. In [10], a Privacy preserving light weight two party protocol is proposed and implemented. However, it is not generalised for multiparty computation.

We propose a multiparty distributed algorithm for privacy-preserving back propagation training with vertically partitioned data1 (i.e., when each party has a subset of features). Our contributions can be summarized as follows.

1) Our paper is the first to investigate the problem of training multi-layered neural networks over vertically partitioned databases with privacy constraints.
2) Our algorithms are provably private in the semi honest model [10] and lightweight in terms of computational efficiency.
3) Our algorithms include a solution to a challenging technical problem, namely, privacy-preserving computation of activation function. This problem is highly challenging because most of the frequently used activation functions are infinite and continuous while cryptographic tools are defined in finite fields. To overcome this difficulty, we propose the first cryptographic method to securely compute sigmoid function for multiparty computation, in which an existing piecewise linear approximation of the sigmoid function [11] is used.

II. TECHNICAL PRELIMINARIES

For ease of presentation, in this paper, we consider a neural network of three layers, where the hidden-layer activation function is sigmoid and the output layer is linear. Note that it is trivial to extend our work to more layers.

1) Semi honest Model: As many existing privacy-preserving data mining algorithms, we adopt semi honest model in this paper. Semi honest model is a standard adversary model in cryptography. In this paper, the security of our algorithm is guaranteed in this model. When computing function in a distributed fashion, semi honest model requires that each party that participates in the computation follow the algorithm, but a party may try to learn additional information by analysing the messages that she receives during the execution. In order to guarantee the security of distributed algorithm of computing, it must be ensured that each party can learn nothing beyond what can be implied by her own input and output. Semi honest model is a right fit for our setting, because normally participants want to learn the neural network learning results and thus they are willing to follow the algorithm to guarantee the results correctness. The security guaranteed in semi honest model can relieve the concerns about their data privacy. Of course, in reality, there may be scenarios in which there are malicious adversaries. It has been shown that a distributed algorithm that is secure in the semi honest model can be converted to one that is secure in the malicious model, with some additional costs in computation and communications for zero knowledge proofs.

2) El Gamal Scheme

El Gamal is a public-key encryption scheme .

Setup

- Choose Large Prime p
- Choose primitive element $\alpha \varepsilon Z_p^*$
- Choose secret key a ε {2,3,…..,p-2}.
- Compute $\beta = \alpha^a \bmod p$.
- Public Key = K_{pub} = (p, α, β).
- Private Key = K_{pr} = (a).

Encryption

- Choose k ε {2,3,…..,p-2}.
- Y1= αk mod p.
- Y2= x.βk mod p
- Encryption: Ekpub (x, k) = (Y1, Y2).

Decryption

- x= Dkpr(Y1, Y2) = Y2 (Y1a)-1 mod p

Homomorphic Property: For two messages m1 and m2, an encryption m1m2 of can be obtained by an operation on E(m1,r) and E(m2,r) without decrypting any of the two encrypted messages.

Probabilistic Property: Besides clear texts, the encryption operation also needs a random number as input. There exist many encryptions for each message.

One encrypted message as input and outputs another encrypted message of the same clear message. This is called re-randomization operation.

3) Piecewise Linear Approximation of Sigmoid Function:

Equation 1 is piecewise linear approximation of sigmoid function $1/1+e^{-x}$ .Our algorithm for BPN Learning makes use of following approximation

$$y(x) = \begin{cases} 1 & x > 8 \\ 0.015628x + 0.875 & 4 < x \le 8 \\ 0.03125x + 0.8125 & 2 < x \le 4 \\ 0.125x + 0.625 & 1 < x \le 2 \\ 0.25x + 0.5 & -1 < x \le 1 \\ 0.125x + 0.375 & -2 < x \le -1 \\ 0.315x + 0.1875 & -4 < x \le -2 \\ 0.015628x + 0.125 & -8 < x \le -4 \\ 0 & x \le -8 \end{cases} \quad \ldots (1)$$

III. PRIVACY PRESERVING NEURAL NETWORK LEARNING

In this section, we present a privacy-preserving distributed algorithm for training the neural networks with back propagation algorithm. A privacy-preserving testing algorithm can be easily derived from the feed forward part of the privacy-preserving training algorithm. Our algorithm is composed of many smaller private computations. We will look into them in detail after first giving an overview

Algorithm No 1:

Statement: Securely Computing the Product of Two Integers.

M= Integer hold by Party A.

N= Integer hold by Party B.

Party A:

1) Generates a Random Number R
2) Computes M.i – R for each I, s.t –n<i<n Mi = M.i – R.
3) Encrypts each M_i using ElGamal Scheme using new random number for each M_i
4) Sends each $E(M_i,r_i)$ to Party B in increasing order of i

Party B:

1) B picks $E(M_N,R_N)$, randomizes it and sends back to A $E(M_N,r')$, $r' = r_N+S$ where S is known to Party B

Party A:

1) Party A partially decrypts E(Mn,r') and sends to B

Party B:

1) Finally decrypts to get Mn = M.N-R

Algorithm No 2:

Statement: Securely Computing Piecewise Linear Sigmoid Two Integers.

M= Integer hold by Party A.

N= Integer hold by Party B.

Party A:

1) Generates a Random Number R
2) Computes y (X1+i)-R for each I, s.t –n<i<n Mi = y (X1+i)-R
3) Encrypts each M_i using ElGamal Scheme using new random number for each M_i
4) Sends each $E(M_i,r_i)$ to Party B in increasing order of i

Party B:

1) B picks $E(M_N,R_N)$, randomizes it and sends back to A $E(M_N,r')$, $r' = r_N+S$ where S is known to Party B

Party A:

1) Party A partially decrypts E(Mn,r') and sends to B

Party B:

1) Finally decrypts to get Mn = y(x1+x2)-R

Algorithm 3: Privacy-Preserving Distributed Algorithm for Back propagation Training

Step1: Feed Forward Stage

For each hidden layer node hj, Party A computes weight * input for Ma attributes

Party B computes weight * input for Mb attributes

Using Algorithm 2, Party A and B jointly compute Sigmoid Function for each hidden layer node hj obtaining their random shares hj1 and hj2 respectively.

Each Party calculates Output Oi respectively.

Step 2: BACK PROAGATION STAGE

Party A Computes Δ1 wij as (oi1-ti)hj1+r11+r21

Party B Computes Δ2 wij as (oi2-ti)hj2+r12+r22

Similarly step is repeated to Hidden Layers to calculate the delta values back propagating from output layer to hidden layer

A sends Δ1 of output and hidden layers to B and B sends Δ2 of output and hidden layers to A.

A and B compute new weight vector values accordingly also considering the learning rate.(At hidden and Output Layers)

This rate is kept same at both parties.

Repeat above three steps until terminating condition for error threshold occurs or after predefined number of iterations.

IV. RESULT ANALYSIS

In this section, we perform experiments to measure the accuracy and overheads of our algorithms. We compare the testing error rates in privacy-preserving and non-privacy-preserving cases. The experimentation was carried on Intel Core I5 Machine with 4GB of memory. The results shown below are the average of 100 runs. The testing data sets are from the University of California at Irvine (UCI) data set repository [12]. We choose a variety of data sets, kr-versus-kp, Iris, Pima-Indian-diabetes (diabetes), Sonar and Landsat with different characteristics, in the number of features, the number of labeled classes, the size of data sets, and data distributions. Different neural network models are chosen for varying data sets. Table I shows the architecture and training parameters used in our neural network model. We choose the number of hidden nodes based on the number of input and output nodes. This choice is based on the criteria of having at least one hidden unit per output, at least one hidden unit for every ten inputs, and five hidden units being a minimum.

Figure 1, 2, 4 and 5 shows different training and testing error rates achieved for back propagation applied for Non Privacy Preserving Algorithm and Sigmoid Approximation Algorithm. The results are taken considering entire datasets for training and also some results are taken with variable datasets for training and testing.

Figure 3 indicates the decrease in the error rates in further increase of number of epochs. This graph only indicates the decrease of error rate for Non Privacy Preserving Algorithm.

Table 1: Architecture and training parameters

	IRIS	KRKP	DIABETES	SONAR
TOTAL SAMPLES	150	3196	768	208
CLASS	3	2	2	2
ARCHITECTURE	4-5-3	36-15-1	8-12-1	60-6-1
EPOCHS	80	20	40	150
LEARNING RATE	0.1	0.1	0.2	0.1

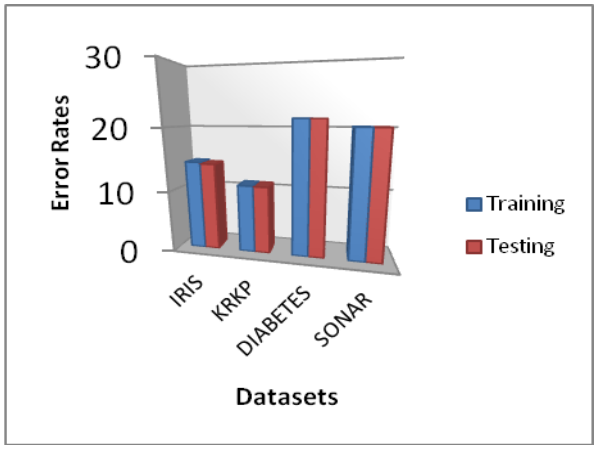

Fig. 1 Non Privacy Preserving Training v/s Testing results (100% Dataset used for Test and Train)

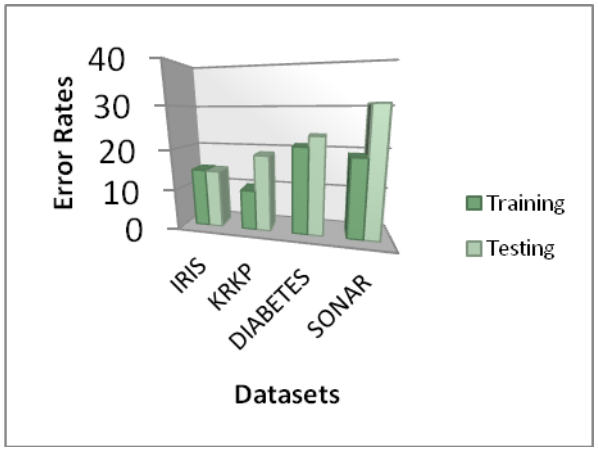

Fig. 2 Non Privacy Preserving Training v/s testing results (Divided Dataset used for Test and Train)

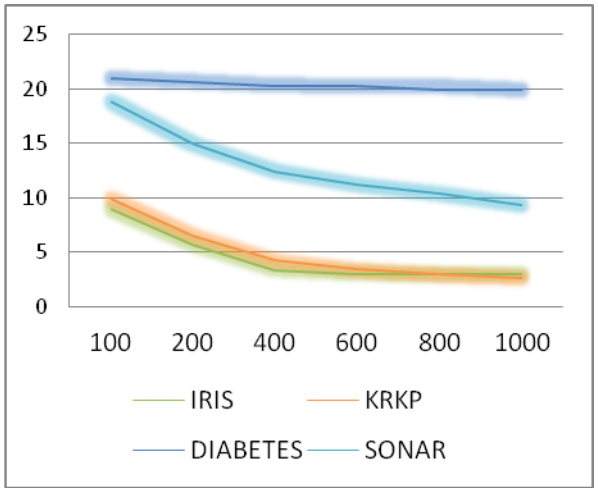

Figure 3: Error Rates on Training Epochs for Non Privacy Preserving Algorithm.

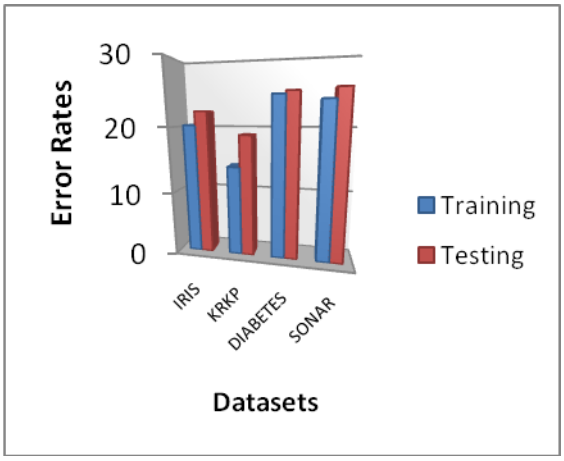

Figure 4: Piecewise Linear Approximation of Sigmoid for 100% dataset.

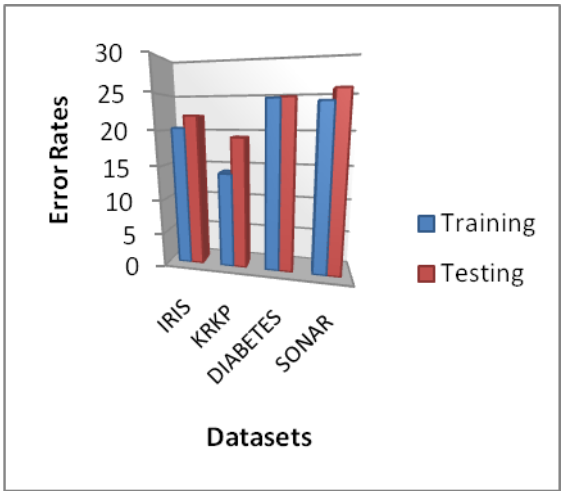

Figure 5: Piecewise Linear Approximation of Sigmoid for variable dataset.

Now we compare the results for Communication versus Computation Overhead for Computation of Product Algorithm Securely.

Product	Computation Time and Percentage		Communication Time and Percentage	
Party A	0.0872	65.07%	0.0468	34.93%
Party B	0.1028	86.82%	0.0156	13.18%
Average	75.93%		24.07%	

Table 2: Computation versus Communication Overhead on Securely Computing Product of Two Numbers.

Analysis based on Time Complexity

In Step 1 of Algorithm 1, there are 2 * n encryptions by party, where n is the parameter in piecewise linear sigmoid function definition. In Step 2, one re-randomization is conducted. In Steps 3 and 4, party A and party B perform one partial decryption, respectively. Total Time: (2n+1) C+2D, where C is cost of encryption and D is cost of partial decryption.

REFERENCES

[1] A.C. Yao, "How to Generate and Exchange Secrets," Proc. IEEE 27th Ann. Symp. Foundations of Computer Science, pp. 162-167, 1986.

[2] W. Du, Y. Han, and S. Chen, "Privacy-Preserving multivariate Statistical Analysis: Linear Regression and Classification," Proc. Fourth SIAM Int'l Conf. Data Mining (SDM), pp. 222-233, Apr. 2004.

[3] W. Du and Z. Zhan, "Building Decision Tree Classifier on Private Data," Proc. IEEE Int'l Conf. Privacy, Security and Data Mining, pp. 1-8, 2002.

[4] S. Han and W.K. Ng, "Privacy-Preserving Linear Fisher Discriminant Analysis," Proc. 12th Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD), May 2008.

[5] Y. Lindell and B. Pinkas, "Privacy Preserving Data Mining," Advances in Cryptology, pp. 36-53, Springer-Verlag, 2000

[6] J. Vaidya and C. Clifton, "Privacy Preserving Association Rule Mining in Vertically Partitioned Data," Proc. Eighth ACM SIGKDD, pp. 639-644, July 2002.

[7] JJ. Vaidya and C. Clifton, "Privacy-Preserving k-Means Clustering over Vertically Partitioned Data," Proc. Ninth ACM SIGKDD, pp. 206-215, 2003.

[8] D. E. Rumelhart, B.Widrow, and M. A. Lehr, "The basic ideas in neural networks," Commun. ACM, vol. 37, pp. 87–92, 1994.

[9] R. P. Lippmann, "An introduction to computing with neural networks," IEEE Acoust. Speech Signal Process. Mag., vol. 4, no. 2, pp. 4–22, Apr.1987.

[10] O. Goldreich, Foundations of Cryptography. Cambridge, U.K.: Cambridge Univ. Press, 2001–2004, vol. 1 and 2.

[11] D. J. Myers and R. A. Hutchinson, "Efficient implementation of piecewise linear activation function for digital VLSI neural networks," Electron.Lett., vol. 25, pp. 1662–1663, 1989.

[12] C. L. Blake and C. J. Merz, UCI Repository of Machine Learning Databases, Univ. California, Dept. Inf. Comput. Sci., Irvine, CA, 1998 [Online].Available: http://www.ics.uci.edu/~mlearn/MLRepository.html

AUTHORS

First Author – Mr. Suhel S Sayyad, B.E degree in Computer Science and Engineering from Shivaji University, Kolhapur. During 2007-2009, he worked with ZenSar Technologies. Currently he is pursuing his M.Tech in Computer Science and Engineering in Walchand College of Engineering, Sangli and working as an Assistant Professor at Annasaheb Dange College of Engineering and Technology, Ashta. Tal: Walwa Dist Sangli

Second Author: Dr. P.J. Kulkarni, PhD. Working as Deputy Director at Walchand College of Engineering, Sangli