International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Privacy Conserving Gradient Descent Method Applied for Neural Network with Distributed Datasets

Sachin P. Yadav[1], Amit B. Chougule[2]

[1] Department of Computer Science and Engineering, D Y Patil College of Engineering, Kolhapur, Maharashtra, India

[2] Professor, Department of Computer Science and Engineering,
Bharati Vidyapeeth's College of Engineering, Kolhapur – 13, Maharashtra, India

Abstract: ***The learning problems have to be concerned about distributed input data, because of gradual expansion of distributed computing environment. It is important to address the privacy concern of each data holder by extending the privacy preservation concept to original learning algorithms, to enhance co-operations in learning. In this project, focus is on protecting the privacy in significant learning model i.e. Multilayer Back Propagation Neural Network using Gradient Descent Methods. For protecting the privacy of the data items (concentration is towards Vertically Partitioned Data and Horizontally Partitioned Data), semi honest model and underlying security of El Gamal Scheme is referred [7].***

Keywords: Cryptography Techniques, Distributed Datasets, Gradient Descent Methods, Back Propagation Neural Network.

1. Introduction

Many techniques in machine learning follow a gradient-descent method in the iterative process of discovering a target functions or decision model. For instance, neural networks generally perform a series of iterations to converge the weight coefficients of edges in the network; thus, settling into a decision model. Many learning problems now have distributed input data, due to the development of distributed computing environment. In such distributed scenarios, privacy concerns often become a big concern. For example, if medical researchers want to apply machine learning to study health care problems, they need to collect the raw data from hospitals and the follow-up information from patients. Then, the privacy of the patients must be protected, according to the privacy rules in Health Insurance Portability and Accountability Act (HIPAA) [1], which establishes the regulations for the use and disclosure of Protected Health Information. Why the researchers would want to build a learning model (e.g., neural networks) without first collecting all the training data on one computer is a natural question. If there is a learner trusted by all the data holders, then the trusted learner can accumulate 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 at present, alliances will greatly benefit the scientific advances [2].

The researchers have the interest to obtain the result of cooperative learning even before they see the data from other parties. As a concrete example, the progress in neuroscience could be boosted by making links between data from labs around the world, but some researchers are reluctant to release their data to be exploited by others because of privacy and security concerns.

2. Literature Review

D. Agrawal and R. Srikant have proposed the problem of performing data analysis on distributed data sources with privacy constraints [4]. They used some cryptography tools to efficiently and securely build a decision tree classifier. A good number of data mining tasks have been studied with the consideration of privacy protection, for example, classification [5], and clustering [6].

In particular, privacy-preserving solutions have been proposed for the following classification algorithms (to name a few): decision trees, naive Bayes classifier [8], and support vector machine (SVM) [9] Generally speaking, the existing works have taken either randomization-based approaches or cryptography- based approaches[7] Randomization-based approaches, by perturbing data, only guarantee a limited degree of privacy.

A.C.Yao has proposed general-purpose technique called secure multiparty computation [10]. The works of secure multiparty computation originate from the solution to the millionaire problem proposed by Yao, in which two millionaires can find out who is richer without revealing the amount of their wealth. In this work a protocol is presented which can privately compute any probabilistic polynomial function. Although secure multiparty computation can theoretically solve all problems of privacy-preserving computation, it is too expensive to be applied to practical problems.

Cryptography-based approaches provide better guarantee on privacy than randomized-based approaches, but most of the cryptography-based approaches are difficult to be applied

with very large databases, because they are resource demanding. For example, although Laur et al. proposed an elegant solution for privacy-preserving SVM in [9], their protocols are based on circuit evaluation, which is considered very costly in practice.

L.Wan, W. K. Ng, S. Han, and V. C. S. Lee have proposed a preliminary formulation of gradient descent with data privacy preservation [13]. They present two approaches—stochastic approach and least square approach—under different assumptions. Four protocols are proposed for the two approaches incorporating various secure building blocks for both horizontally and vertically partitioned data.

3. Problem Definition

The researchers have the interest to obtain the result of cooperative learning of multiple parties' data for solving classification problems, but some researchers are reluctant to release their data to be exploited by others because of privacy and security concerns. Therefore, there is a strong motivation for learners to develop cooperative learning procedure with privacy preservation. Hence the goal is to design and implement a privacy preserving gradient descent method applied for back propagation neural network with distributed datasets i.e. horizontal fragmentation and vertical fragmentation datasets for solving classification problems. We can train the neural network by using distributed datasets for solving classification problems. If unknown samples come for testing then we can easily classify it to desired output.

4. System Design and Implementation

In this paper, focus is to implement the privacy-preserving distributed algorithm to securely compute the piecewise linear function for the neural network training process to obtain the desired output. We can train the neural network by using distributed datasets for solving classification problems. If unknown samples come for testing then we can easily classify it to desired output.

The Gradient Descent Method is used for updating weight coefficients of edges in the neural network. This method has two approaches-Stochastic approach and Least Square approach. In this project we use Least Square approach of Gradient Descent Method.

4.1 System Architecture

Figure 4.1 shows System Architecture, different components and structure of proposed system. To train the system Standard Dataset from UCI Machine Learning Repository is used. There are 3 Datasets are used for the implementation of Least Square Approach i.e. Iris Flower Dataset, Tennis Dataset and Wine Dataset.

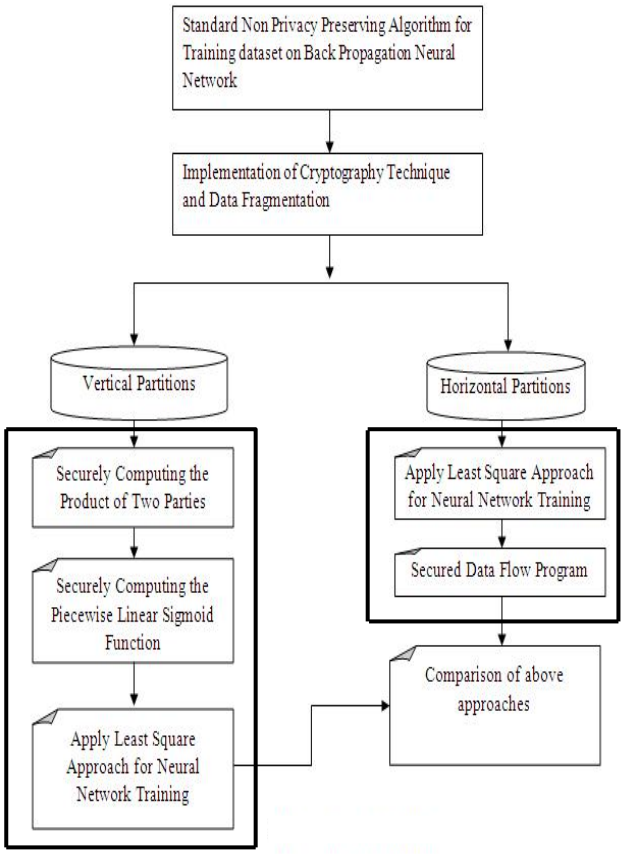

Figure 1: System Architecture

4.1.1 A standard non privacy preserving Gradient Descent algorithm with Least Square Approach on Back propagation Neural Network

Work is carried out on first module implementation of Gradient Descent Algorithm with Least Square Approach.
The back propagation is applied for Least Square Approach where the iteration is decided based on the tolerable error allowed to stop the training process. The literature behind the same and sample hand simulation is shown in the below mentioned figure 2.

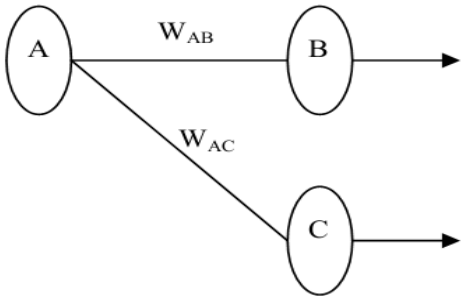

Figure 2: Sample hand simulation of Neural Network

The connection we're interested in is between neuron A (a hidden layer neuron) and neuron B (an output neuron)and has the weight WAB. The figure 4.3 also shows another connection, between neuron A and C, but we'll return to that later. The algorithm works like this:

1. First apply the inputs to the network and work out the output – remember this initial output could be anything, as the initial weights were random numbers.

2. Next work out the error for neuron B. The error is what you want – What you actually get, in other words:
   ErrorB= OutputB(1-OutputB)(TargetB– OutputB).
3. The "Output(1-Output)" term is necessary in the equation because of the Sigmoid Function – if we were only using a threshold neuron it would just be (Target – Output).
4. Change the weight. Let W+AB be the new (trained) weight and WAB be the initial weight.
   W+AB= WAB + (ErrorB x OutputA)
   Notice that it is the output of the connecting neuron (neuron A) we use (not B). We update all the weights in the output layer in this way.
5. Calculate the Errors for the hidden layer neurons. Unlike the output layer we can't calculate these directly (because we don't have a Target), so we Back Propagate them from the output layer (hence the name of the algorithm). This is done by taking the Errors from the output neurons and running them back through the weights to get the hidden layer errors. For example if neuron A is connected as shown to B and C then we take the errors from B and C to generate an error for A.
   ErrorA= OutputA(1 - OutputA)(ErrorBWAB + ErrorCWAC)
   Again, the factor "Output (1 - Output)" is present because of the sigmoid squashing function.
6. Having obtained the Error for the hidden layer neurons now proceed as in stage 3 to change the hidden layer weights. By repeating this method we can train a network of any number of layers.

4.1.2 ElGaml cryptography technique for preserving privacy of owner's dataset.

Work is carried out on second module Implementation of ElGamal Encryption cryptography technique for preserving privacy of owner's dataset.

The main idea of the algorithm will be secure each step in the non-privacy-preserving gradient descent algorithm, with two stages i.e feed forward and back propagation. In each step, neither the input data from the other party nor the intermediate results can be revealed. Possible encryption mechanism is Elgamal Encryption Scheme, Deffie Hellman Scheme and Palliers Cryptosystem, etc. The work will make use of Elgamal encryption scheme.

ElGamal encryption Algorithm:

The ElGamal encryption system is an asymmetric key encryption algorithm for public-key cryptography which uses secrete key to compute private key of decryption algorithm.

ElGamal encryption consists of three components: the key generator, the encryption algorithm, and the decryption algorithm.

1. Key generator

i. Generate large prime p and generator g of the multiplicative Group $\mathbb{Z}_p^*$ of the integers modulo p.
ii. Select a random integer a, 1 <= a<= p- 2, and compute ga mod p.
iii. Public key is (p; g; ga); secrete key is a.

2. Encryption

i. Obtain public key (p, g, ga).
ii. Represent the message as integers m in the range {0,1,….,p-1}.
iii. Select a random integer k, 1<=k<=p-2.
iv. Compute γ=gk mod p and δ = m*(ga)k.
v. Send ciphertext c=(γ, δ).

3. Decryption

i. Use secret key a to compute (γp-1-a) mod p.
ii. Recover m by computing (γa) * δ mod p.

4.1.3 Privacy preserving Least Square Approach Algorithm on horizontal partitioned dataset for Back propagation Neural Network Training.

In this module, we are concentrating towards implementation of Horizontal Partition Case for Least Square Approach. This Horizontal Partition Case is experimented for two parties. Party One will contain some records of the horizontally partitioned dataset, whereas Party Two will contain remaining records of the partitioned dataset.

1. Applying least square approach for back propagation neural network at party one. It will generate the output weight vectors. These weight vectors are stored in .csv file format.
2. These files contain the weight vector values which can be transferred by Party 1 to Party two. These weight vector values will not disclose any sort of input to the other party and hence preserves the privacy of the data elements.
3. Party two further performs his training by reading this partial trained weight vector values using least square approach for back propagation neural network and the final generated weight vector values can be used by both parties for performing testing.

4.1.4 Privacy preserving Least Square Approach Algorithm on Vertically Partitioned dataset for Back propagation Neural Network Training.

1. Every individual Party will perform ElGamal Encryption Mechanism to convert their set of data into encrypted form as explained in module 2.
2. All encrypted data is collected at a single repository.
3. Normalization of the record is made in the range of -1 to 1
   For this we use following equation
   In = (I-min) ((newMax-newMin)/(Max-Min)) +newMin
   In our case newMin is -1 and newMax = +1
4. Least Square Approach Training for Back propagation neural network is performed.

For better understanding, the back propagation learning algorithm can be divided into two phases: propagation and weight update.

Phase 1: Propagation

Each propagation involves the following steps:

1. Forward propagation of a training pattern's input through the neural network in order to generate the propagation's output activations.
2. Backward propagation of the propagation's output activations through the neural network using the training pattern's target in order to generate the deltas of all output and hidden neurons.

Phase 2: Weight update

For each weight-synapse follow the following steps:

1. Multiply its output delta and input activation to get the gradient of the weight.
2. Bring the weight in the opposite direction of the gradient by subtracting a ratio of it from the weight.

This ratio influences the speed and quality of learning; it is called the learning rate. The sign of the gradient of a weight indicates where the error is increasing; this is why the weight must be updated in the opposite direction.
Repeat phase 1 and 2 until the performance of the network is satisfactory.

4.1.5 Comparison Study and Analysis.
In this module, we analyses the accuracy factor between Non Privacy Preserving Least Square Approach, Privacy Preserving Least Square Approach on Horizontal Dataset and Privacy Preserving Least Square Approach on Vertical Dataset. Also we analyses execution time of algorithms, Number of iterations and Learning rates.

5. Experimental Result and Analysis

Several standard datasets are available for experimental purpose which is taken from UCI Machine Learning Repository. 3 Datasets are used for the implementation of Least Square Approach of Gradient Descent Method on Back Propagation Neural Network. The dataset used for experimentation are Iris, Tennis and Wine.
The project work is carried out by using Java language with version Jdk1.6 on Windows 7 operating system.

5.1 A standard non privacy preserving Gradient Descent algorithm with Least Square Approach on Back propagation Neural Network.

In this module we set Execution Parameters for the datasets as shown in Table 1

Table 1: Execution Parameters for the datasets

Sr. No	Dataset	Number of Hidden Units	Layer Structure	Learning Rate	Momentum	Number of Iterations
1	Iris	2	4-2-3	0.25	0.1	450
2	Tennis	2	4-2-1	0.25	0.1	450
3	Wine	10	13-10-3	0.25	0.1	5000

The Results of the Testing Accuracy for the above mentioned parameters is as shown in Table 2

Table 2: Testing Accuracy for the datasets

Sr. No	Name of Dataset	Number of Test Samples	Correct Predictions	Incorrect Prediction	Percentage Accuracy
1	Iris	50	49	1	98%
2	Tennis	4	2	2	50%
3	Wine	13	9	4	69.23%

Figure 3 shows the graph of testing accuracy of non privacy preserving gradient descent method applied on different datasets.

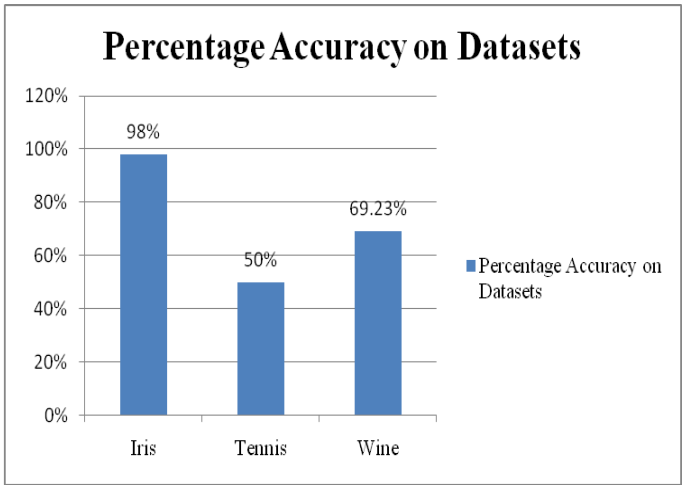

Figure 3: Percentage Accuracy on Datasets

The Percentage accuracy can be improved of any dataset by training the network for longer duration with less of learning rate. This behavior can be analyzed using the statistics as shown in Table 3

Name of Dataset: Tennis

Table 3: Improvement of percentage accuracy of Tennis dataset

Learning Rate	Number of Iterations	Percentage Accuracy	Remarks
0.25	450	50	Sample 1
0.1	450	75	Improvement in test results due to reduction of learning rate against Sample 1
0.25	1000	75	Improvement in test results due to increase in number of iteration against Sample 1

5.2 Privacy preserving Least Square Approach Algorithm on horizontal and vertical partitioned dataset for Back propagation Neural Network Training.

We analyses the accuracy factor between Non Privacy Preserving Least Square Approach, Privacy Preserving Least Square Approach on Horizontal Dataset and Privacy Preserving Least Square Approach on Vertical Dataset. Table 4 shows the comparison of percentage accuracy of testing datasets of three approaches.

Table 4 Testing Accuracy of datasets

		NonPrivacy Preserving Algorithm			Privacy Preserving Horizontal Algorithm for Least Square Approach			Privacy Preserving Vertical Algorithm for Least Square Approach		
Name of Data set	Number of Test Samples	Correct Predictions	InCorrect Predictions	Accuracy	Correct Predictions	InCorrect Predictions	Accuracy	Correct Predictions	InCorrect Predictions	Accuracy
Iris	50	49	1	98%	47	2	94%	42	8	84%
Wine	13	9	4	69.23%	9	4	69.23%	9	.4	69.23%
Tennis	4	2	2	50%	2	2	50%	N.A	N.A	N.A

The Tennis dataset is a character based dataset. The ElGamal Security scheme works with numeric data type. So in above Table 4 there is no result shown for vertical partitioned datasets. Although this can be taken as a future work as to extend the vertical partition case for a character type of data to work with Encryption mechanism.

Figure 4 shows the graphical representation of comparisons of accuracy of testing datasets for three approaches.

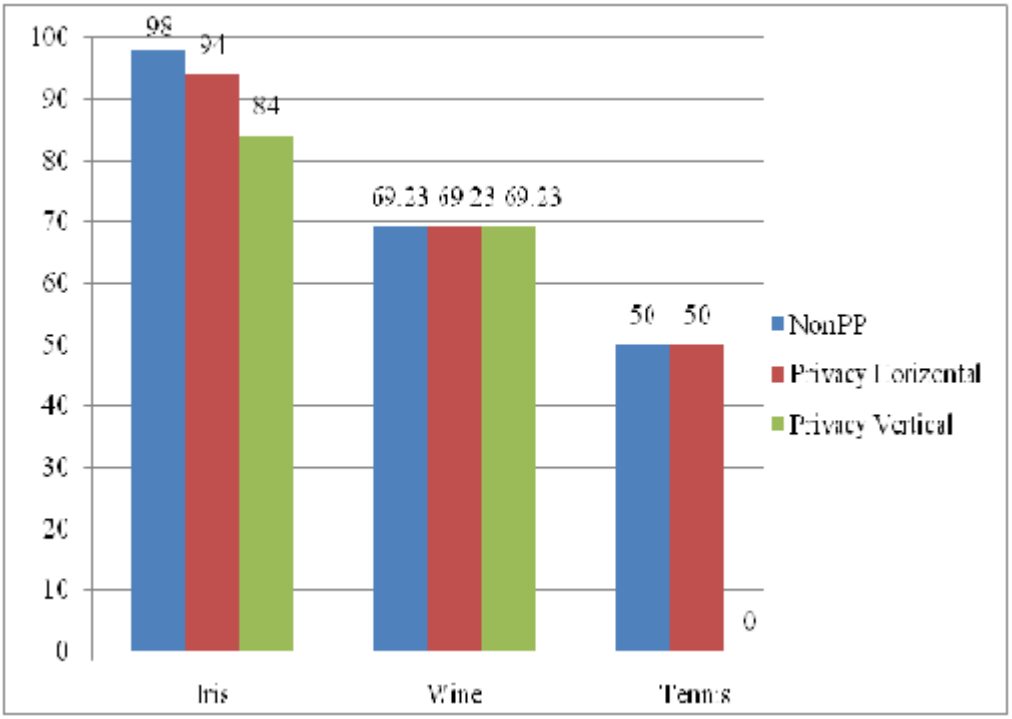

Figure 4: Graphical representation of comparisons of accuracy of testing datasets for three approaches

From Figure 4 it is observed that there is no significant loss in the accuracy if training is performed on ample number of iterations. The number of iterations and the learning rate are modified for the Privacy Preserving Vertical Partition Case as the cryptographic implementation makes the data loss to happen. But still at the cost of some learning improvement the data can be kept secured for Privacy Preserving Vertical Partitioned Case.

6. Conclusion

Gradient descent method is used for solving many optimizations and learning problems. In this paper, we presented a secure gradient descent method for training of neural network for distributed datasets. Gradient Descent method contains two approaches i.e. stochastic approach and Least Square approach. For our work least square approach is used and it is well suited for training of neural network. We use Back Propagation neural network for training purpose.
We works on privacy preserving protocols for securely performing gradient descent method over vertically or horizontally partitioned data based on the least square approach between two trusted parties. Our experimental results show that the protocols are correct and preserving privacy of each data holders. We also conducted experiments to analyze the results of non privacy preserving least square approach with privacy preserving least square approaches for horizontal and vertical portioned datasets. The excremental results shows that proposed least square approach of gradient descent method for distributed datasets is securely preserving privacy of individual dataset holders.

6.1 Future Work

For future work, we will extend our work for distributed datasets i.e. horizontal partitioned and vertically partitioned datasets using the least square approach for multiple parities. Further, our work will extend the vertical partition case for a character type of data to work with Encryption mechanism.

References

[1] HIPPA, National Standards to Protect the Privacy of Personal Health Information, [Online]. Available: http://www.hhs.gov/ocr/hipaa/finalreg.html
[2] M. Chicurel, "Data basing the brain," Nature, vol. 406, pp. 822–825, Aug. 2000.
[3] D. Agrawal and R. Srikant, "Privacy-preserving data mining," in Proc. ACM SIGMOD
[4] Y. Lindell and B. Pinkas, "Privacy preserving data mining," in Lecture Notes in Computer Science. Berlin, Germany: Springer-Verlag, 2000, vol. 1880, pp. 36–44.
[5] N. Zhang, S. Wang, and W. Zhao, "A new scheme on privacy-preserving data classification," in Proc. ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining, 2005.
[6] G. Jagannathan and R. N. Wright, "Privacy-preserving distributed k-means clustering over arbitrarily partitioned data," in Proc. ACM
[7] O. Goldreich, Foundations of Cryptography. Cambridge Univ. Press, 2001.
[8] R. Wright and Z. Yang, "Privacy-preserving Bayesian network structure computation on distributed heterogeneous data," in Proc. 10th ACM SIGKDD.
[9] H. Yu, X. Jiang, and J. Vaidya, "Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data," in Proc. Annu. ACM Symp. Appl. Comput., 2006.
[10] A. C. Yao, "Protocols for secure computations," in Proc. 23rd Annu. Symp. Found. Comput. Sci., Chicago, IL, Nov. 1982.
[11] M. Barni, C. Orlandi, and A. Piva, "A privacy-preserving protocol for neural-network-based computation," in Proc. 8th Workshop Multimedia Security, New York, 2006.
[12] A. Yao, "How to generate and exchange secrets," in Proc. 27th IEEE Symp. Found. Comput. Sci., 1986, pp. 162–167.
[13] L.Wan, W. K. Ng, S. Han, and V. C. S. Lee, "Privacy-preservation for gradient descent methods," in Proc. IEEE Transactions on Knowlede and Data Engineering, 2010.

Author Profile

Mr. S. P. Yadav , BE degree in Information Technology from Shivaji University Kolhpur,Maharashtra,India.Currently he is pursuing his ME in Computer Science and Engineering in D.Y.Patil College of Engineering and Technolgy,Kolhapur,Maharashtra and working as a Assistant Professor at Annasaheb Dange college of engineering,Ashta,Tal:Walwa,Dist Sangli.

Prof. A. B. Chougule, M.Tech. in Computer Science and Technology from Shivaji University Kolhapur, Maharashtra, India. Working as Professor and Head of Computer Science and engineering Department at Bharati Vidyapeet's College of Engineering, Kolhapur, Maharashtra, India.