Preserving Privacy Using Gradient Descent Methods Applied for Neural Network with Distributed Datasets

Mr.Sachin P.Yadav, Mr.Amit B.Chougule

**[1] D.Y.Patil College of Engineering
Kolhapur,Maharashtra,India**

**[2] Bharati Vidyapeet's College of Engineering,
Kolhapur, 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, Neural Network.*

1. Introduction

Many techniques in data mining and machine learning follow a gradient-descent paradigm 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. Related Work

2.1 "Privacy-Preserving Data Mining"

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].

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.

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.

2.2 “Protocols for Secure Computations”

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.

2.3 “Privacy-Preserving Gradient-Descent Methods”

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.

Major headings are to be column centered in a bold font without underline. They need be numbered. "2. Headings and Footnotes" at the top of this paragraph is a major heading.

3. Gradient Decent Method

Gradient descent is a general paradigm that underlies many algorithms in machine learning and knowledge discovery. In neural networks, updating the weight value of the output and hidden nodes is a form of gradient descent. There are two approaches of Gradient Decent Method-Stochastic Approach and the Least Square Approach.

For proposed system Least Square Approach is suitable for Back Propagation Neural Network.

3.1 Stochastic Approach:

For neural networks, the two component functions for the prediction function f in unipolar sigmoid activation function in the multilayer perceptron are: $\mathbf{hj}(\mathbf{xj}, \mathbf{wj}) = \mathbf{wjxj}$ and $\mathbf{gh} = \dfrac{\mathbf{1}}{\mathbf{1} + \mathbf{e}^{-\alpha \mathbf{h}}}$ where α is a positive constant.

Correspondingly, the bipolar sigmoid activation function is $\mathbf{g}(\mathbf{h}) = \dfrac{\mathbf{1} - \mathbf{e}^{-\alpha \mathbf{h}}}{\mathbf{1} + \mathbf{e}^{-\alpha \mathbf{h}}}$ and $\mathbf{hj}(\mathbf{xj}, \mathbf{wj}) = \mathbf{wjxj}$.

3.2 Least Square Approach:

The objective of the gradient descent is to determine w to best fit the training data that minimize the error function. In the following, we introduce a simpler case that can be computed based on the least square approach. Here, we assume that the global error function is the Residual Sum of Squared (RSS) values in (1) for the training data.

$$\mathbf{RSS} = \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \left(\mathbf{yi} - \mathbf{f}(\mathbf{xi})\right)^{\mathbf{2}} \tag{1}$$

Besides, we define the prediction function f as a composition of two functions g and h where 1) function g is an invertible function (such as the inverse function of $\mathbf{y} = \dfrac{\mathbf{1}}{\mathbf{1} + \mathbf{e}^{-\alpha \mathbf{x}}}$ is $\mathbf{x} = -\dfrac{\mathbf{1}}{\alpha} \ln\left(\dfrac{\mathbf{1}}{\mathbf{y}} - \mathbf{1}\right)$ and 2) function h is a linearly separable function:

$$\mathbf{h}(\mathbf{xi}) = \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{m}} \mathbf{xi}, \mathbf{jwj} \tag{2}$$

That is applied by many gradient-descent methods.

4. Comments and Need of Work

From the above survey it can be comment that:

1. The Gradient Descent methods are used for solving optimization problems. This method gives general formulation for preserving privacy of distributed datasets to solve optimization problems.

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.

2. The privacy preserving Gradient Descent Method can be applied for some specific areas like Neural Network, Linear Regression, and Bayesian Network.

3. There is no any privacy preserving mechanism for distributed datasets (Vertically partitioned and horizontally partitioned) using Gradient Descent Method in neural network for solving classification problem.

The work carried out in above Literature Survey is based on privacy preserving Data Mining for performing data analysis on distributed datasets. Further the solution proposed in the Privacy Preserving Gradient Descent Methods gives general formulation for preserving privacy of distributed datasets to solve optimization problems. It is therefore required that this solution for the privacy preserving gradient descent method needs to be converted to the Neural Network for solving classification problem.

5. Proposed Work

Here 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.
Briefly the work can be summarized into following proposed system architecture.

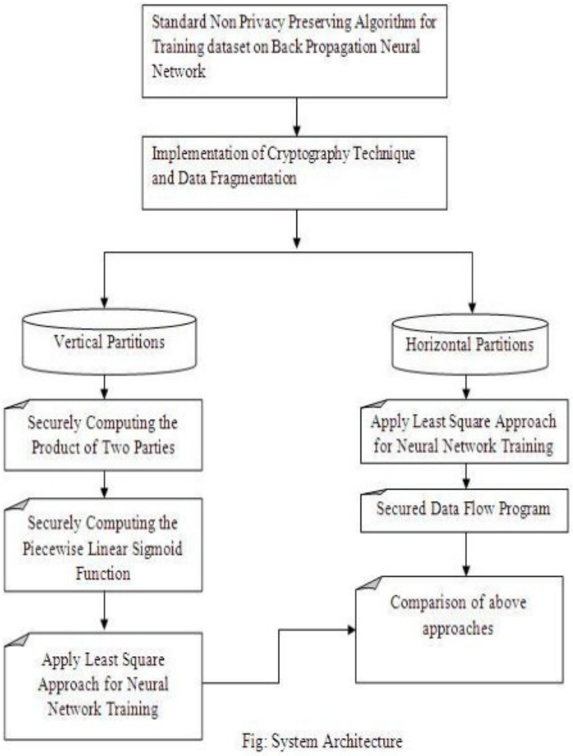
Standard Non Privacy Preserving Algorithm for Training dataset on Back Propagation Neural Network
Implementation of Cryptography Technique and Data Fragmentation
Vertical Partitions
Horizontal Partitions
Securely Computing the Product of Two Parties
Apply Least Square Approach for Neural Network Training
Secured Data Flow Program
Securely Computing the Piecewise Linear Sigmoid Function
Comparison of above approaches
Apply Least Square Approach for Neural Network Training

Fig: System Architecture

The Proposed system will contain following modules.

5.1 Implementing a standard non privacy preserving algorithm on Neural Network

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 net-work 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.

3) 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.

4) Repeat phase 1 and 2 until the performance of the network is satisfactory.

There are two modes of learning to choose from, one is on-line (incremental) learning and the other is batch learning. In on-line (incremental) learning, each propagation is followed immediately by a weight update. In batch learning, much propagation occurs before weight updating occurs. Batch learning requires more memory capacity, but on-line learning requires more updates.

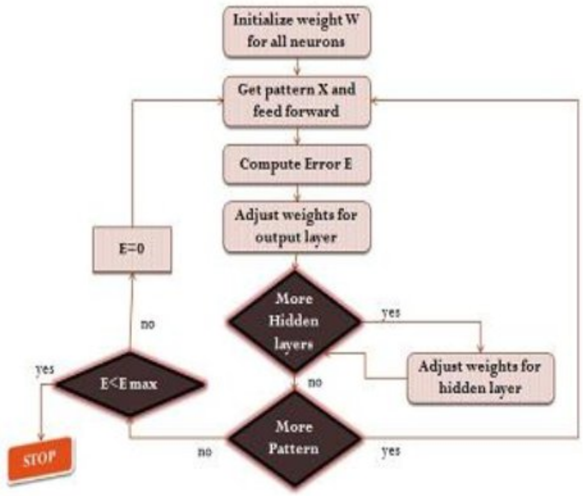
Initialize weight W for all neurons
Get pattern X and feed forward
Compute Error E
Adjust weights for output layer
E=0
More Hidden layers
yes
no
Adjust weights for hidden layer
yes
E<E max
no
STOP
More Pattern
no
yes

Fig: Data Flow of a standard non privacy preserving algorithm on Neural Network

5.2 Implementing cryptography technique for preserving privacy of owner’s dataset and fragmentation of data

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.

a. Semi honest Model:

As many existing privacy-preserving data mining algorithms, here in this work semi honest model is adopted. Semi honest model is a standard adversary model in cryptography. In this work, the security of 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 analyzing 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 t for this work's 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.

We use here El-Gamal Cryptography technique.

- El Gamal Encryption Scheme:
- El Gamal is a public-key encryption scheme.
- Setup
    - o Choose Large Prime p
    - o Choose primitive element αεZp*
    - o Choose secret key a ε {2,3,…..,p-2}.
    - o Compute β = αa mod p.
    - o Public Key = Kpub = (p, α, β).
    - o Private Key = Kpr = (a).
- Encryption
    - o Choose k ε {2,3,…..,p-2}.
    - o Y1= αk mod p.
    - o Y2= x.βk mod p
    - o Encryption :Ekpub (x, k) = (Y1, Y2).
- Decryption
    - o x= Dkpr(Y1, Y2) = Y2 (Y1a)-1 mod p

1. 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.

2. 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

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.

message of the same clear message. This is called re-randomization operation.

5.3 Implementation of Securely Computing the Piecewise Linear Sigmoid Function for Vertically partitioned dataset.

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:

Securely Computing the Product of Two Integers.

Assume 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 Mi using ElGamal Scheme using new random number for each Mi
4) Sends each E(Mi,ri) to Party B in increasing order of i

Party B:

1) B picks E(MN,RN), randomizes it and sends back to A E(MN,r’), r’ = rN+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:

Securely Computing Piecewise Linear Sigmoid Function

Assume 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 Mi using ElGamal Scheme using new random number for each Mi
4) Sends each E(Mi,ri) to Party B in increasing order of i

Party B:

1) B picks E(MN,RN), randomizes it and sends back to A E(MN,r’), r’ = rN+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

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

Algorithm: Least Square Approach for Vertically Partitioned case for Two Parties.

Initialization to Random weight values and making them known to both parties
And for all Training Sample Repeat below steps
Step1: Feed Forward Stage

1) For each hidden layer node hj, Party A computes weight * input for Ma attributes
2) Party B computes weight * input for Mb attributes
3) 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.
4) For each output layer oi , Party A computes oi1as

$$o_{i1} = \sum_{i} w_{ij} {}^{\circ} h_{j1}$$

5) For each output layer oi , Party B computes oi2 as

$$o_{i2} = \sum_{i} w_{ij} {}^{\circ} h_{j2}$$

Step 2: Back Propagation Stage

1) For each Output Layer weight , parties A and B apply Algorithm 1 to securely compute product $h_{j1}o_{i2}$ obtaining random shares r11 and r12. Similarly they compute $h_{j2}o_{i1}$, to get r21 and r22 as shares

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

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

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

Step 3:

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

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

3) This rate is kept same at both parties.

4) Finally, repeat above three steps until terminating condition for error threshold occurs or after predefined number of iterations.

5.5 We can apply Least Square approach algorithm for horizontal partitioned dataset for effectively obtaining exact output for classification of data.

Algorithm: Least Square Approach for Horizontally partitioned case for Two Party
Two rounds of BPA algorithm should be called and for a single round of algorithm logic is same as Non privacy preserving algorithm for BPA training. This training suits the Least Square Approach.

6. Conclusions

Using Privacy preserving gradient decent method applied for Back Propagation Neural network on distributed datasets. We can preserve privacy of dataset holders and no one can get others private data then also our neural network is gets trained for distributed datasets.

We can extend this work for other types of neural network in future. And also generalize the neural network.

Acknowledgments

I thanks to Prof. G.A.Patil Head of Department Computer Science and Engineering, D Y Patil College of Engineering, Kolhapur. And I also thanks to Prof. Amit B.Chougule for supporting me.

References

[1] HIPPA, National Standards to Protect the Privacy of Personal HealthInformation,[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 Sym p. 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.

First Author: 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.

Second Author: Prof. A. B. Chougule, M.Tech.Working as Professor at Bharati Vidyapeet's College of Engineering, Kolhapur, Maharashtra, India.

Copyright (c) 2013 International Journal of Computer Science Issues. All Rights Reserved.