International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Privacy Preserving Protocol for Two-Party Classifier Over Vertically Partitioned Dataset Using ANN

Smitha Iddalgave[1], Sumana M[2]

[1]IV semester M.Tech, Information Science & Engg, M S Ramaiah Institute of Technology, VTU University
MSR Nagar, MSRIT Post, Bangalore -54, India
[2]Professor, Department of Information Science & Engg, M S Ramaiah Institute of Technology, VTU University
MSR Nagar, MSRIT Post, Bangalore -54, India

Abstract: ***With the emergence of distributed computing privacy preservation has become a priority concern. Privacy in the field of data mining can be ensured by having secure computations. Data mining in distributed scenario deals with data from multiple data providers. The providers have to be assured about the safety of their data. Hence, rather than having a trusted party (network) which can collect data from providers and perform meaning classification on the combined data, we propose a two party classifier which allows the network participants to work on their dataset and communicate with each other in a secure manner using encryption schemes to establish relation between their data without revealing any of their private data to each other. The participants can only learn about their input and output values. The protocol is implemented on vertically partitioned dataset, as with horizontal partitioning the processing becomes sequential with the output of one network being fed to other for further processing. This protocol can be extended for multiple participants and checked for its privacy-preservation property.***

Keywords: Neural network, backpropagation, learning, privacy-preserving, cryptographic scheme.

3. Introduction

Neural networks, also known as Artificial Neural networks (ANN) are nothing but computational models which consist of multiple simple processing units (artificial neurons) which communicate with each other via multiple weighted connections. These networks are adaptive in nature i.e. they learn by example. This feature of ANN helps in application domains where the data is readily available but has incomplete or little understanding of problem. ANNs are useful in identifying and learning correlated patterns between input data and their corresponding target values. After learning, they can be used to predict the outcome of new independent input data [19].

Generally, a series of iterations are performed by neural networks to congregate the weight coefficients of connections in the network; thus, settling into a decision model/ classifier. Until distributed mining emerged, the network would get its input data from a single data owner or the network owner itself has data to train the network. In such cases, there is no concern about preserving data privacy. However, as the distributed computing started blooming it became necessary to take care of privacy concerns of input data wherein there are many data providers available in order to train the network, but would be taken aback due to their concern about their proprietary data security and privacy. Hence, secure multiparty computation and privacy preservation has grabbed the attention of many researchers to come up with machine learning algorithms and as a result many works have been done in recent past to fight this concern [1][2][3][4] & [5].

3.1. Existing System

An approach is proposed in [6] concerning privacy preservation using gradient descent method. However, the protocol takes into account a single data owner and the protocol works on a network topology with no hidden layer and one node in the output layer.

Another privacy preserving approach [7] deals with a neural network with the owner owning only the network without any data. The aim of this approach is to ensure that the owner of network does not obtain any knowledge about the data of data providers, and at the same time also ensuring that the providers do not gain any knowledge within the network.

Both these approaches deal with privacy preserving, but in different context.

3.2. Proposed System

This paper proposes two-party classifier algorithm for distributed data scenario. Its features are as stated below

1. The network learns over a vertically partitioned data inputs using artificial neural network. The learning technique followed is supervised backpropagation with preserving the privacy of data owners.
2. The algorithm is semihonest in nature i.e. each party learns from its own input and output without acquiring any knowledge about other participant's data.
3. Since privacy is a major objective here, it needs to use certain cryptographic tools to maintain secure computation of activation function. However, this task is more challenging as most of the cryptographic tools are defined on finite field whereas the activation functions are infinite in nature.

3.3. Objectives

1. Knowledge Gain.
2. Collaborate multiple data owners.
3. Maintain privacy with distributed data sources.
4. Co-operative learning on combined without revealing details of input or intermediate results.
5. Establishing meaningful relation between distributed data.

3.4. Challenges

1. Network deals with multiple data owners i.e. working on vertically partitioned dataset.
2. Maintaining privacy of oneself's data being without revealed to other owners trying to learn during learning process.
3. Applying cryptographic tools defined in finite field on activation function.

4. Literature Survey

4.1. Privacy preserving secure computations

The notion of privacy preserving computations came up with the solution proposed by Yao in [8] for millionaire problem where the millionaires find the richest among them without revealing their wealth details. However, such solutions cannot be applied in reality as they are resource demanding and hence are costly in practical. The main intention of privacy preserving computation is to keep the computation secure such that at the end no party can learn anything other than his/her input data and final results. This can be achieved by having a trusted party who can take the data, perform computations and return results to participants. However, even with trusted party there can questions from the data provider's side with respect to their data security. A solution to this would be to let the participants communicate with each other and perform the computations collectively to obtain global results. However, the communication can be made secure by letting non-determinism in the intermediate values exchanged between the parties. This is done by using cryptographic tools to encrypt the values before sending to other party which can then obtain a predicted intermediate result that is likely to actual value [9]. Clifton et al. [10] lists various tools of privacy preserving distributed computation techniques towards an attempt to build a toolkit of all proposed techniques till date.

4.2. Towards Privacy Preserving Data Mining

In [11] Y Lindell and Benny Pinkas discuss about how secure computation can be applied to privacy preserving data mining and the problems that arise while setting privacy-preserving data mining. Also they provide a survey of the highly efficient techniques proposed till date along with the most common errors in the literature of privacy preserving data mining.

A two party protocol proposed by Y Lindell and Benny Pinkas [4] works on how privacy preserving data mining can be performed when the data is distributed at two or more parties who jointly perform data mining operations on the combined data and learn global results without revealing data at their individual sites. Both the parties jointly build a decision-tree without either of the party learning any information other than the global results.

4.3. Randomization Technique

In the past many privacy-preserving solutions have been proposed for different classification algorithms. R Agarwal and Ramakrishnan S [12] present an approach to data mining with privacy by distorting or perturbing the sensitive attributes in the input dataset before providing to the mining process. Different randomization functions, such as Gaussian or uniform perturbations, of modifying the values are presented along with measures to be able to re-construct the original data from the randomized data.

On contrary, the approaches (such as the ones specified in [13], [14] [15]) using cryptography seem to provide better guarantee in terms of privacy than the prior approach i.e., randomized-based approaches.

4.4. Privacy Preserving two party protocol for Gradient Descent Methods.

Most of the current privacy preservation approaches firstly isolate execution and verify for the correctness of the results of some of the basic operations of the algorithm, as ensuring secure computation of these operations can assure that the algorithm as a whole is also secure. In [16] Li Wan and Wee Keong also make use of the same ideology in their protocol implementation. They propose a protocol on a vertically partitioned dataset and demonstrate its execution for privacy preserving classification in gradient descent methods using artificial neural networks. The protocol is implemented with two parties using one of the secure scalar product protocol presented by Du and Atallah in [17]. Gradient descent paradigm triggers techniques in data mining and machine learning such as updation of weights for hidden and output nodes in neural networks.

4.5. Cryptographic Methods

In paper [18], TaherElGamal presents a study about ElGamal signature scheme with respect to other cryptographic schemes such as RSA, Diffie-Hellman Key Distribution. According to this scheme, if there are p messages then every message m from p can have a lot many signatures, but every single signature obtained belongs to only one of the messages. The re-randomization property of ElGamal produces different cipher texts for the same message(plain text) each time the encryption operation is performed on plain text, such that there will not be any relation among the different cipher texts obtained. This prevents cipher attacks like probable text attack where even if the intruder suspects what the plain text could be and tries to encipher it again, then the new cipher text will be different from the current cipher text and hence the intruder can never be sure of what the original plain text is. Also the cipher text obtained in ElGamal is double the size of cipher text in RSA, which makes the ElGamal system difficult to break[18]. Hence, providing higher security.

4.6. Privacy Preserving using cryptographic scheme

Barni et al. [7] proposed algorithm for privacy preserving of weight vectors and activation functions. These algorithms were proposed for training a neural network which does not own any input data for training the network. The purpose of proposing the algorithms was to ensure that the neural network does not learn any critical data of the input data received from different data providers. Also, care was taken not to reveal the input data of the data providers to one another. These assumptions vary from the scenario in which the algorithms proposed in this paper. Hence, protocols in [7] are not suitable for our experiment.

4.7. RapidMiner

The data from the distributed sources may be of infinite range. Hence, it is required to normalize the data values to fit into a specific range accurate processing. Rapidminer is a tool which provides an integrated environment for various functionalities. It is enabled with a user friendly graphical interface to promote design and execution of analytical workflows called processes. Each process consists of multiple "Operators". The operators within each process are desired to perform specific tasks. The normalize operator takes as input the output of the retrieve operator. The Meta data containing the attributes of dataset has to be attached along with data. The range within which the original data has to be fit can be specified as a parameter to normalize operator.

5. Motivation & Problem Statement

5.1. Motivation

With the emergence of distributed computing, to obtain more of realistic data for analyzing the scenarios in an efficient way, managing data from various data owners has become crucial issue. With distributed data mining it is important to consider the need of maintaining privacy of data from different sources. The motivation leading to the implementation of our algorithm is to present a classifier built by multiple data owners for establishing relation between their data to classify the combined data in a more meaningful manner, without revealing their data from the dataset or intermediate results during computation.

This section briefs data normalization, backpropagation network algorithm employed, the approximation technique used for activation function and the cryptographic tool used.

5.2. Preliminaries

5.2.1. Data Normalization

One of the crucial preprocessing steps is input data normalization. Normalizing the input data within a defined range significantly improves the performance of multilayer neural networks. To use the data as inputs of the neural network, scaling or normalization should be realized for each attribute.

5.2.2. Backpropagation Learning

The training approach used here is supervised backpropagation learning. It does learning on multi-layer neural networks by iteratively processing the inputs to learn the weights for predicting the class label of tuples. The working of this method can be showed in 2 phases:

Phase 1: Propagation

It includes both the forward and backward propagation

1. Forward propagation: of the weighted input tuples through the neural network to arrive at the results at output node via the activation function at hidden nodes.
2. Backward propagation: of output results through the neural network along with arrive at new weights by computing the deltas of all output and hidden nodes/neurons.

Phase 2: Weight update

For each weight-synapse follow the following steps:

1. Obtain delta weight for input and output weights
2. Subtract a ratio (percentage) of the delta weight from the initial weights of input and output nodes.

This ratio (percentage) is called the learning rate, which has great impact on the speed of learning process. With higher learning rates the speed of learning increases, but more accurate training can be achieved with lower learning rates. The sign of the delta weights indicates where the error is increasing. The above phases are repeated until the network performance is satisfactory.

5.2.3. Cryptographic Tool

ElGamal is the cryptographic tool utilized in the algorithm. It is a public-key encryption scheme defined on any cyclic group.

a) Properties

In the privacy-preserving algorithms for neural network learning, a homomorphic cryptographic scheme of ElGamal is utilized for the following 2 properties:

1. Homomorphic Property

ElGamal scheme is homomorphic, which means an encryption operation on m_1m_2 can be performed by performing encryption on E (m_1, R) and E (m_2, R) without having to decrypt either of the cipher texts.

2. Probabilistic Property

ElGamal scheme is also probabilistic, which means that the encryption operations requires a random number as input along with the plain text. A single plaintext can be encrypted to many possible cipher texts. It mainly consists of the following 3 phases:

1) Key Generation phase: Generate a cyclic group G of order 'q' with generator g.

1. Choose random number x from g i.e. {1…., q-1}
2. Compute h = gx
3. Publish h along with description of public key (G, q, g). x is retained as it is private key .

2) Encryption Phase: Encrypts a message 'm' using public key (G, q, g, h)

1. Choose random number y from {1,……q-1}
2. Compute c1=gy
3. Calculate shared secret s = hy
4. Convert secret message 'm' into an element m` of G
5. Calculate cipher text c2 = m`. s
6. Send cipher text (c1, c2) = (gy, m`. hy) = (gy, m`.(gx)y)

A new y generated for every message to improve security. Hence y is called "ephemeral key".

3) Decryption Phase: To decrypt a cipher text (c1, c2) with private key x.

1. Calculate shared secret s = c1x
2. Compute secret message m`
3. Calculate m` = c2. s-1 and convert to plain text message m. where s-1 is inverse of sin the group G.

b) Semihonest Model

The semihonest model is a standard adversary model in cryptographic approach.. In a distributed scenario working with semihonest model, while computing certain function f, it is required that each party involved in the computation of the function must follow the algorithm and may try to learn some additional information by analyzing the data it receives from other participants. This model ensures that none of the party is able to learn anything more than what is implied from its input and output. This requirement ensures that the data from the other participants are not revealed during the combined computation and guarantees complete privacy preserving.

The reason why this model is suitable in this scenario is, since here the participants need to know the results of the network learning but would not like to reveal their own internal data to the other participant and therefore they agree to follow the algorithm to guarantee the correctness of results while relieving from the concerns of data privacy. The problem can be stated as below based on the semihonest model:

5.3. Problem Statement

Suppose that a set of training samples are vertically partitioned between two parties A and B such that party A holds a data set D1 with mA attributes and party B Holds a data set D2 with mB attributes for each data entry. Let each tuple in D1 be denoted as x_A, where $x_A = \{x_1, x_2, x_3,\ldots,x_{mA}\}$ and each tuple in D2 be denoted as x_B, where $x_B = \{ x_{m+1}, x_{m+2}, x_{m+3}\ldots\ldots x_{mA+mB}\}$.

Privacy-preserving two-party classifier neural network training is that, during each round of training/learning both the parties together compute the additive values of the weights of the connections without revealing their input data to each other.

Having the training samples x_A from party A and x_B from party B, the goal here is to let each party obtain its own share of additive value of weight Δw without the other party knowing its input data i.e., party A does not know x_B and party B does not know x_A.

To summarize, the paper is mainly considers the privacy concerns only to the insiders i.e. concerns of the participants of learning process and it is expected that all the participants have appropriate access controls and security communication techniques. In our work, both the parties have a set of attributes and share the class label. They work collaboratively by exchanging their random shares generated as intermediate results. The new weights are calculated by using the learning factor jointly decided by both the participants.

6. Privacy-Preserving 2-party Classifier

The proposed algorithm focuses on preserving privacy between the two participating parties which train the artificial neural network using backpropagation learning technique.

6.1. Training Algorithm:

During every iteration in the training process, the inputs are the values held by the 2 parties i.e., x_A by party A and x_B by party B and the target vector for the current training data (t_x). The target vector is made known to the participating parties. The connected weights of the output and hidden layer will be the output of the algorithm.

Input: $(\{x_{A,} x_{B}\}, \{t_x\})$

Output : $\{w_{ij}^{o}, w_{jk}^{h} \mid \forall_k \in\{1,2,3,\ldots..,a\}, \forall_j \in\{1,2,3,\ldots.,b\}, \forall_i \in\{1,2,3,\ldots.,c\}\}$

This algorithm utilizes the feedforward and backpropagation stages to secure every step of the nonprivacy-preserving backpropagation algorithm. Any party will not be aware of the input data or intermediate results of the other party during each step. This is accomplished a party by randomly sharing each of its result with the other party such that neither of the parties can imply the original data information from the intermediate results. The phrase "randomly sharing" here refers to the random numbers held by each party such that the sum of their random numbers would equal the intermediate result.

Once the training without compromising privacy is complete, both the parties collectively establish a new neural network depicting the properties of the union dataset.

Following are the algorithms needed for the implementation of privacy preserving two-party classifier over vertically partitioned dataset.

Algorithm 1: Privacy-Preserving Algorithm for Back propagation Training over vertically partitioned datasets.

Consider party A has m_A attributes and party B has remaining attributes of the input dataset.

Initialize weights to some random number and broadcast to both the parties.

Step1: Feedforward Stage

1. For each node h_j, in the hidden layer
   a. Summation of weight (from input to hidden node) * input for m_A attributes and summation of weight (from input to hidden node) * input for mB attributes is computed by party A and party B respectively.
   b. Sigmoid Function using algorithm 2 to obtain random shares h_{j1} (random share of party A) and h_{j2} (random share of party B) is collaboratively computed by both the parties respectively.
2. For each output layer node o_i,
   a. Party A computes summation of weight *(from hidden to output node)* $* h_{j1} = o_{i1}$.
   b. Party A computes summation of weight *(from hidden to output node)* $* h_{j2} = o_{i2}$.

Step 2: Backpropagation Stage

1. For each node O_i in output layer,
   a. The delta weights are computed by both parties as below
      i. $\Delta_1 w_{ij} = (o_{i1}-t_i)\, h_{j1}+r_{11}+r_{21}$.
      ii. $\Delta_2 w_{ij} = (o_{i2}-t_i)\, h_{j2}+r_{12}+r_{22}$.

   Where t_i is the target class value and r_{11}, r_{12}, r_{21} and r_{22} are the random shares obtained by party A and party B by applying algorithm 3 on $h_{j1}o_{i2}$ and $h_{j2}o_{i1}$ respectively, such that $h_{j1}o_{i2} = r_{11} + r_{12}$ and $h_{j2}o_{i1} = r_{12} + r_{22}$.
2. For each output layer node,

a. Party A and B jointly compute their random shares q_1 and q_2 using algorithm 4.
b. Applying algorithm 3 on their input attribute and random shares q_1 & q_2 respectively party A and B obtain their random shares r_{61} & r_{62}.
c. Both parties obtain their shares of delta weights i.e. Δw_1 & Δw_2.

3. Party A and party B exchange Δw_1 & Δw_2 of output and hidden layers with each other.
4. A and B compute the new weights considering a ratio of delta weights. The ratio (learning rate) is decided collaboratively by both the parties.

The above steps are repeated until the number of epochs or till error threshold has been reached. All the tuples from the training dataset are processed using this algorithm.

Step 3: End

Algorithm 2: Securely Computing Piecewise Linear Sigmoid Two Integers.

Assumptions:

M = Integer held by Party A.
N = Integer held by Party B.

Step 1: Party A performs the following operations

1. Generate Random Number R.
2. For each *i*, $y\ (x_1+i) - R$ is computed, such that $-n < i < n$
3. Define $m_i = y\ (x_1+i) - R$.
4. Each m_i is encrypted using ElGamal scheme with new random number for each m_i.
5. The set of $E\ (m_i, r_i)$ are sent to Party B in the order of i.

Step 2: Party B performs the following operations

1. Picks $E\ (m_N, r_N)$ among the set of $E\ (m_i, r_i)$ corresponding to its input.
2. Re-randomizes $E\ (m_N, r_N)$.
3. Sends $E(m_N, r')$ back to A, where $r' = r_N + s$ and s is known to Party B

Step 3: Party A performs the following operations

1. Send partially decrypted $E(m_N, r')$ to B.

Step 4: Party B performs the following operations

1. Obtains $m_n = y(x_1+x_2) - R$ by further decrypting the partial decrypted value received from party A.

Step 5: End

Algorithm 3: Securely Computing the Product of Two Integers.

Assumptions:

M= Integer held by Party A.
N= Integer held by Party B.

Step 1: Party A performs the following operations

1. Generate Random Number R.
2. For each *I*, compute $M.i - R$ such that $-n < i < n$.
3. Defines $m_i = M.i - R$.
4. Each m_i is encrypted with ElGamal scheme using new random number r_i for each m_i.
5. Set of $E\ (m_i, r_i)$ is sent to Party B in rising order of *i*.

Step 2: Party B:

1. Picks $E\ (m_N, r_N)$ from the set of $E\ (Mi, ri)$ corresponding to its input.
2. Re-randomize $E\ (m_N, r_N)$.
3. Sends $E\ (m_N, r')$ back to A, where $r' = r_N + s$ and s is known to only Party B.

Step 3: Party A:

1. Send partially decrypted $E\ (m_N, r')$ to B.

Step 4: Party B:

1. Finally B decrypts the partially decrypted message from A to get $m_N = M.N-R$.

Step 5: End

Algorithm 3: Securely computing random share q_1 and q_2

Input: h_{j1}, O_{i1} and h_{j2}, O_{i2} for Party A and Party B respectively from algorithm 1.

Output: Random shares q_1 and q_2 for A and B respectively.

Step1: Both parties obtain random shares r_{31} & r_{32} using algorithm 3 on h_{j1} h_{j2}.

Step2: Compute p_1, s_1 for party A and p_2, s_2 for party B as below

a. $p_1 = h_{j1} - h^2_{j1} - 2\ r_{31}$
b. $p_2 = h_{j2} - h^2_{j2} - 2\ r_{32}$
c. $s_1 = \sum_i (-t_i + O_{i1}) * weight\ (to\ output\ node)$
d. $s_1 = \sum_i O_{i2} * weight\ (to\ output\ node)$

Step3: Party A obtains random shares r_{41} & r_{51} and Party B obtains random shares r_{42} & r_{52} using algorithm 3 on s_1 p_2 and s_2 p_1 respectively.

Step4: Compute q_1, q_2 locally as below

a. $q_1 = s_1p_1 + r_{41} + r_{51}$
b. $q_2 = s_2p_2 + r_{42} + r_{52}$

Step5: End

7. Evaluation

7.1. Setup

All the listed algorithms are implemented in java using NetBeans IDE. The experiment was carried out on a Windows XP workstation with dual 1.6GHz Intel processors and 2GB RAM. The experiment is carried out by varying the number of epochs and learning rate for the network. Both the parties collaboratively decide upon the learning rates. Table I shows the architecture of the considered network and the training parameters considered for carrying out experiment.

Based on input and output nodes, the number of hidden nodes can be selected i.e. the number hidden nodes can be increased to obtain better accuracy. The criterion used is to have at least one hidden node per output value. The attribute values of the dataset at both parties are normalized to fit within range of 0-1.

Table 1: Dataset, architecture and training parameters

Dataset name	*Tuple*	*Class Label*	*Architecture*	*Epoch*	*Learning Rate*
Cancer dataset	100	2	7-2-1	30	0.05
				60	0.05
				100	0.05
				30	0.07
				60	0.07
				100	0.07
				30	0.1
				60	0.1
				100	0.1

7.2. Computation speed analysis

The following results were obtained with the experiment carried out for learning rate, computation speed and number of epochs.

Firstly, the result of increasing the number of epochs is analyzed when the network is trained with a fixed learning rate (the graph is for learning rate 0.1). The number of epochs can be decided based on the architecture and number of samples in the dataset. Higher the number of samples less should be the epoch. However for analysis purpose we have considered lower epochs even for lesser samples. The experiment is carried out for 30, 60 and 100 epochs on the same dataset. The results are plotted in graph1 shown in figure.

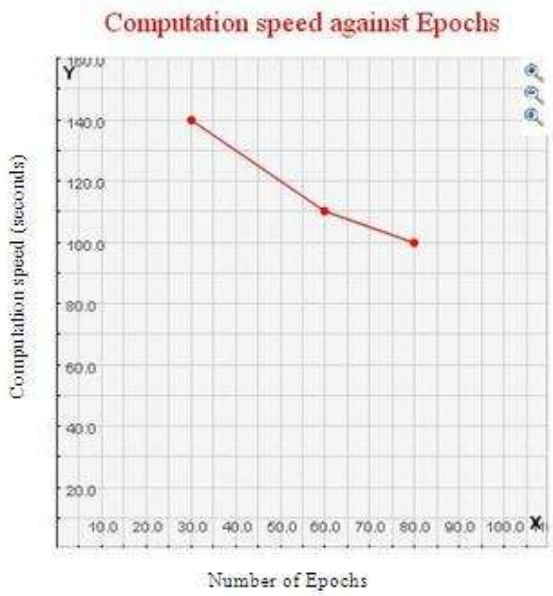

Figure 1: Graph for demonstrating the effect of number of Epochs on computation speed.

It can be seen from the graph that with the increase in the number of epochs the computation or the learning process gradually speeds up.

Secondly, the result for varying learning rate is analyzed while keeping the number of epochs fixed to 100. The learning rate can also be used to analyze the accuracy of the classifier. From the graph from figure 2 it can be seen that with lesser learning the learning process is more accurate.

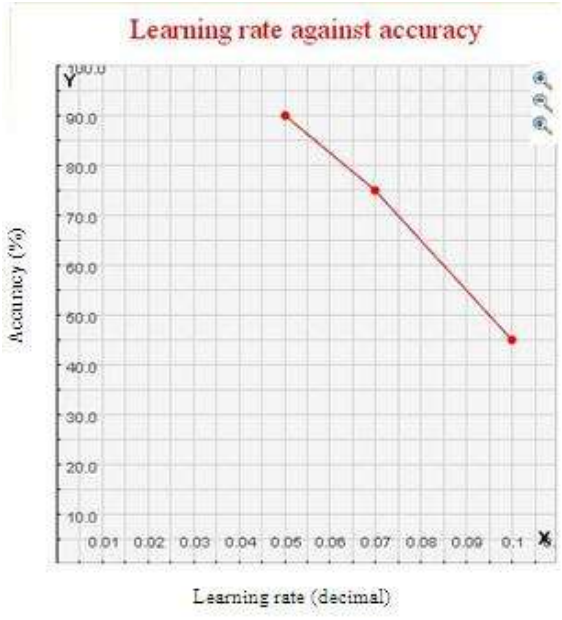

Figure 2: Graph for demonstrating the effect of learning rate on accuracy.

Thirdly, the effect on computation speed is analyzed with varying learning rates. The number of epochs is fixed to 100. From the graph in figure 3 it can be seen that with higher learning rates the computation speeds up.

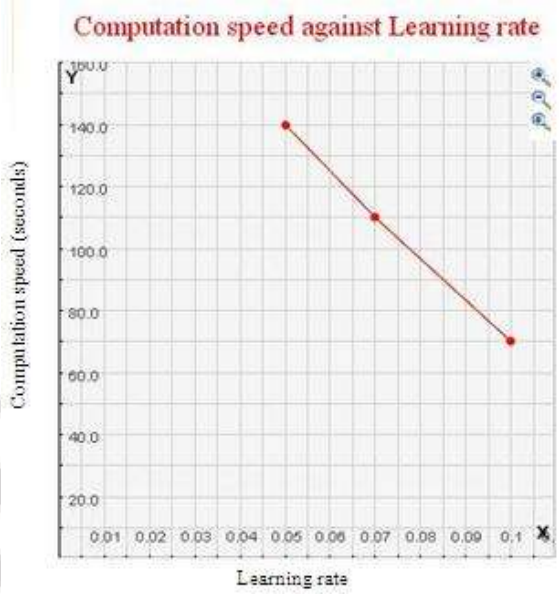

Figure 3: Graph for demonstrating the effect of learning rate on computation speed.

8. Conclusion and Future scope

8.1. Conclusion

A new approach on preserving privacy of distributed data for training the neural network has been proposed. The protocol designed helps to maintain privacy of the network participants working collaboratively on vertically partitioned datasets. The work mainly explains how the parties providing distributed data can jointly train the newly built neural network with their combined data without revealing any of their intricate details, such as their input data and intermediate results generated during the learning process, to one another. Thus it is possible to establish the relation between the data from different providers and classify the combined data in a more meaningful way using the proposed protocol.

8.2. Future Scope

The work is performed on two parties jointly working to establish a new neural network for the combined data. This can be extended to multiple parties using either backpropagation learning or other neural network learning techniques. Also, the work can be modified to be more secure by letting the parties train the next round using only their random shares of weights. Also different encryption schemes can be used to speed up the computations.

References

[1] W.Du, Y. Han, and S. Chen, "Privacy Preserving multivariant statistical analysis: Linear regression and

Classification", Proc. Fourth SIAM Int'l Conf.Data Minign(SDM), pp222-233, April 2004.
[2] 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.
[3] 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.
[4] Y. Lindell and B. Pinkas, "Privacy Preserving Data Mining," Advances in Cryptology, pp. 36-53, Springer-Verlag, 2000.
[5] J. Vaidya and C. Clifton, "Privacy Preserving Association Rule Mining in Vertically Partitioned Data," Proc. Eighth ACM SIGKDD, pp. 639-644, July 2002.
[6] L.Wan, W. K. Ng, S. Han, and V. C. S. Lee, "Privacy-preservation for gradient descent methods," in Proc. ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining, 2007, pp. 775–783.
[7] 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, pp. 146–51.
[8] A. Yao, "How to generate and exchange secrtes," in Proc. 27th IEEE Symp. Found. Comput. Sci., 1986, pp. 162–167.
[9] O. Goldreich, S. Micali, and Wigderson. Foundations of Cryptography. Cambridge, U.K.: Cambridge Univ. Press, 2001–2004, vol. 1 and 2.
[10] Chris clifton, Jaideep Vaidya, "Privacy-Preserving Data Mining: Why, How, and When" November-December 2004 (vol. 2 no. 6) pp. 19-27.
[11] Yehuda Lindell¤ and Benny Pinkasy, "Secure Multiparty Computation for Privacy-Preserving Data Mining" in The Journal of Privacy and Con¯dentiality (2009) 1, Number 1, pp. 59-98.
[12] R Agarwal and Ramakrishna S, "Privacy-Preserving Data Mining".
[13] M. Chicurel, "Databasing the brain," Nature, vol. 406, pp. 822–825, Aug. 2000.
[14] Editorial, "Whose scans are they, anyway?," Nature, vol. 406, p. 443,Aug. 2000.
[15] D. E. Rumelhart, B.Widrow, and M. A. Lehr, "The basic ideas in neural networks," Commun. ACM, vol. 37, pp. 87–92, 1994.
[16] L.Wan, W. K. Ng, S. Han, and V. C. S. Lee, "Privacy-preservation for gradient descent methods," in Proc. ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining, 2007, pp. 775–783.
[17] W. Du and M. Atallah. Privacy-Preserving Cooperative Statistical Analysis. Proceedings of the 17th Annual Computer Security Applications Conference, pp 103-110, Louisiana, USA, 2001.
[18] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," IEEE Trans. Inf. Theory, vol. IT-31, no. 4, pp. 469–472, Jul. 1985.
[19] A. Kumar, "Artificial Neural Networks for Data Mining", http://www.iasri.res.in/sscnars/content_dm.htm.

Author Profile

Smitha Iddalgave received B.E degree in Information Science and Engineering from Gogte Institute of Technology, VTU in 2009. She is currently pursuing her M.Tech in software engineering from MSRIT, VTU. She has selected "Privacy preserving data mining using ANN" as a part of M.Tech dissertation.

Sumana M received B.E. and M.Tech degrees in Computer Science and engineering from Manipal Institute of Technology and VTU University Karnataka, respectively and is pursuing Ph.D in computer science and Engineering from the Manipal University. She is presently working as an Assistant Professor in the department of Information Science and Engineering. Her current research interests include data mining, cryptography and privacy preservation. She is a Life Member of the Indian Society for Technical Education (ISTE), the System Society of India.