International Journal of Technical Research and Applications e-ISSN: 2320-8163,
www.ijtra.com Volume 3, Issue 2 (Mar-Apr 2015), PP. 52-55

PRIVACY PRESERVING BACK-PROPAGATION NEURAL NETWORK LEARNING USING CLOUD COMPUTING

Mr. Anwar Basha. H, Mrs. Valli N
S.A. Engineering College
Chennai, India
Anwar.mtech@gmail.com, nvalliamb@gmail.com

***Abstract*—The learning in neural network takes place with the help of Back Propagation Algorithm. The more the number of datasets, the more improved the learning will be. To facilitate the learning process to take place in a secure manner, the cloud platform is used. The cloud distributes the keys to the owners, the participating entities in the learning process. The owners encrypt their data using AES cryptography and then the cloud manipulates the ciphertext using BGN homomorphic algorithm. The collaborative learning takes place in cloud enabling the owners to share the data and then train the neural network with different datasets. Thus the scalability of the learning process is improved.**

Keywords—Privacy preserving, learning, neural network, back-propagation, cloud computing, computation outsource, homomorphic encryption

I. INTRODUCTION

The back propagation algorithm is an efficient technique for learning in neural network. It has got many applications such as stock market prediction, innumerable medical applications and pattern matching. The main challenge in back propagation algorithm is that it has to sufficiently train the neural network with many input datasets. How to provide the neural network with more datasets. The answer is distributed environment. However the drawback with this method is that it is restricted to two party scenario. To extend this learning to multiparty scenario, we avail the cloud computing services.

The cloud computing relies on shared resources available in the Internet rather than the local area network system. The backbone of cloud computing is virtualization. Virtualisation refers to creating virtual memory, virtual operating system, virtual network or a virtual server. Virualization is different from emulation is that it does not try to imitate any other device.

The major issue faced by the network environment is providing the privacy for the data owned by the participants. The new paradigm that is emerging is cloud computing security that addresses this problem. The virtualization in cloud computing security adds to the existing security problems.

In multiparty scenario, multi layered neural network is the preferred one. But that has a potential drawback since the number of hidden layers in the neural network will increase and hence there will be an increase in complexity.

There are three main challenges in handling neural network learning: 1) The privacy of the data owned by the owner that is the data input to the neural network, the intermediate outputs and activation function. 2) the cost associated with the communication and coordination among the multiple parties 3) The method of partitioning the datasets that is vertically and horizontally partitioning the datasets in an arbitrary manner.

Neural Network is the collection of nodes and edges. The edges are associated with a weight and there is an activation function associated with the network that takes weights as inputs and generates the output. The activation function is either a step function or a sigmoid function. Sigmoid function is used in the proposed solution.

[13] Sigmoid Function: Unlike the step function, which is a linear function, the Sigmoid function a non-linear function, and is best suited for Backpropagation. The function is given by the following expression:

$$f(Net) = \frac{1}{1 + e^{-Net_i}}$$

where Net is the input and f is the activation function.s

The graph for the Sigmoid function is

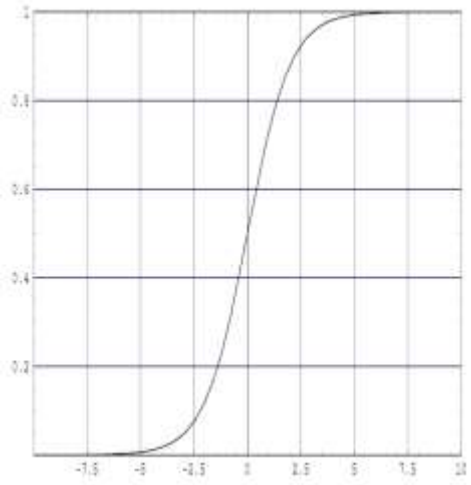

Fig 1. Sigmoid grph

II. SYSTEM MODEL

The system consists of three major components:

- Trust Agent
- Cloud
- Owners

The Owner is the entity which uploads data and participates in learning in the Cloud. The Owner registers with the Cloud. The Cloud performs key generation and distributes the keys to the Owners. After registration the Trust Agent should authenticate the Owner. Then the account of Owner will be activated. The Owner then logs in and uploads data in the Cloud during the learning process, if he/she has data otherwise he/she can take part in the learning.

The datasets are arbitrarily partitioned using horizontal and vertical partitioning. The partitioning of data is done to

improve scalability of learning and availability of the system. Then the partitioned dataset is encrypted using the AES symmetric encryption algorithm. The Cloud performs homomorphic encryption using BGN (Boneh, Goh and Nissim) doubly homomorphic algorithm on the encrypted data using scalar product and scalar sum.

The homomorphic encrypted data is transferred from Cloud to the Owner. Owner performs Back Propagation Neural Network Learning. After the learning, the output weights are updated in the Database (in homomorphic encrypted form). Then the sharing of datasets take place in collaborative learning. The privacy of the Owner dataset is preserved using homomorphic encryption.

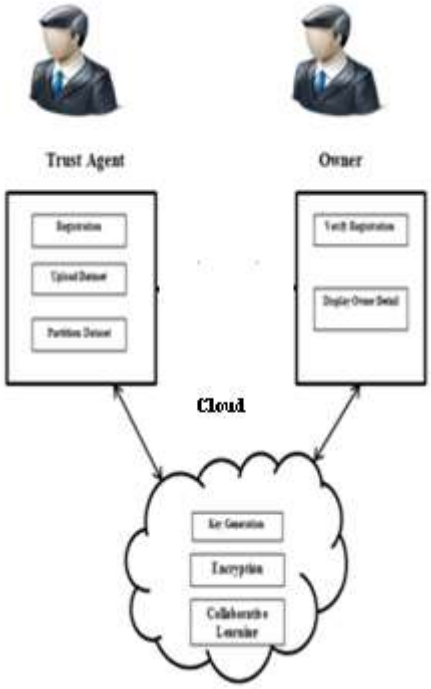

Fig 2. Overall architecture

III. SECURITY MODEL

The Trust Agent is usually a government agency that authenticates the owner by verifying the owner details. In the proposed solution, the cloud generates the keys and distributes to the owners. The security of the system is improved as the cloud does not know the data of the owners.

IV. SYSTEM DESCRIPTION

A. Owner Authentication

The Trust Agent authenticates the owner. Only after the owner is authenticated, the owner can actively participate in the learning process.

B. Uploading the data

The owner uploads the data in cloud. The owner can upload as many times as required. Owner owns a private data set and wants to perform collaborative learning. For this he loads the encrypted data in cloud. Each participating party s, denoted as Ps, owns a private data set and wants to perform collaborative BPN network learning with all other participating parties. That is, they will collaboratively conduct learning over the arbitrarily partitioned data set, which is private and cannot be disclosed during the whole learning process. We assume that each participating party stays online with broadband access to the cloud and is equipped with one or several contemporary computers, which can work in parallel if there is more than one.

C. Arbitrary partitioning of data

The horizontal and vertical combination of dataset is partitioned. The datasets are either vertically partitioned or horizontally partitioned. depending on their configuration and storage requirements. For some datasets Suppose D is the data to be partitioned then D_i is the partition of data D such that $D_1 \cap D_2 \cap D_3 \cap \ldots\ldots\ldots\ldots D_n = \emptyset$
and $D1 \bigcap D_2 \bigcap D_3 \bigcap \ldots\ldots\ldots\ldots D_n = D$

D. Data Encryption

The Data is encrypted using AES cryptography. AES stands for Advanced Encryption Standard. It was designed by Rijmen-Daemen in Belgium. It has 128/192/256 bit keys and 128 bit data. It has design simplicity and resistance against known attacks.

E. Homomorphic encryption of data

BGN (Boneh, Goh and Nissim algorithm is used to encrypt the cipher text and the encrypted cipher text is available in cloud. It is called doubly homomorphic encryption. Homomorphic encryption enables operations on plaintexts to be performed on their respective cipher texts without disclosing the plaintexts. Most existing homomorphic encryption schemes only support single operation - either addition or multiplication. It introduced a public-key 'doubly homomorphic encryption scheme (called 'BGN' for short), which simultaneously supports one multiplication and unlimited number of addition operations. Therefore, given cipher texts C(m1), C(m2), …. C(mi) and C(ˆ m1), C(ˆm2),….C(ˆmi), one can compute C(m1mˆ1+m2mˆ2+ • • • +mimˆi) without knowing the plaintext, where C() is the cipher text of message mi or mˆi, encrypted by the system's public key.

Input: Ciphertext of ϵ
Output: Shares of ϵ: ϵ_s for $P_s, 1 \leq s \leq Z$
begin
  for $s = 1, 2 \cdots, Z$ **do**
    Choose $L_s \xleftarrow{R} (0, u)$
    $C(L_s) = g_1^{L_s} h_1^{r_s n^2}$
    //where u is the upper bound of ϵ
  //Cloud Calculates:
  $C(sumL) = \prod_{s=1}^{Z} C(L_s)$
  case $1.\epsilon > \sum_{s=1}^{Z} L_s$
    $C(\hat{L}) = C(\epsilon) * C(sumL)^{-1}$
  case $2.\epsilon < \sum_{s=1}^{Z} L_s$
    $C(\hat{L}) = C(sumL) * C(\epsilon)^{-1}$
  Decrypt $C(\hat{L})$ with Algorithm 3 and send $\hat{L}$ to P_1
  //Output Shares:
  $\epsilon_1 = L_1 + \hat{L}$ (Case 1) or $\epsilon_1 = L_1 - \hat{L}$ (Case 2)
  for $i = 2, 3 \cdots, Z$ **do**
    $\epsilon_s = L_s$
end

Figure. 2 Scalar Product and Sum

Fig 3[12] illustrates the calculation of scalar product and Sum. The cloud calculates first the product of cipher text and then it calculates the sum of the product obtained in the previous step. The C(m1) is multiplied with C(m^1) and then added with C(m2) multiplied with C(m^2).

F. Back Propagation Learning

Back propagation, otherwise known as "backward propagation of errors", is a common method of learning in artificial neural networks. From a desired output, the network learns from many inputs Back-Propagation neural

network learning algorithm is mainly composed of two stages: feed forward and error back − propagation. In Feed Forward the network is trained with many inputs and the output is generated as a sigmoid function of the inputs. The actual output and the expected output is compared and then the difference is calculated as error. The error is then propagated backwards in the network and the internal weights adjusted so that the difference between the actual and expected weight diminishes.

The input and output [10] of the neuron, i, (except for the input layer) in a multilayer perceptron mode, according to the Back Propagation algorithm are:

$$\text{Input } x_i=\sum w_{ij}o_j+b_i \quad (1)$$

$$\text{Output } o_i=f(x_i) \quad (2)$$

Where W_{ij} is the weight of the connection from neuron i to node j, b_i is the numerical value and f is the activation function.

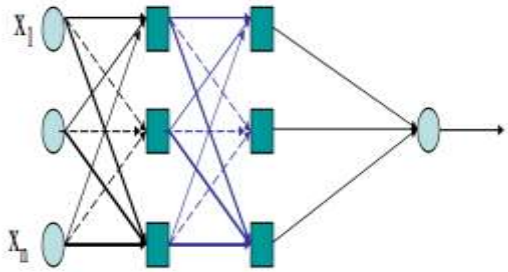

Fig. 3 Configuration of Back Propagation Network

Fig.1[14] shows is a configuration for a three layered Back Propagation network. The inputs (x1..xn) are feeded to the input layer. The weights are assigned to the edges connecting these layers and the output is calculated with the help of sigmoid function. The difference between the actual and the desired output is calculated and this difference is observed as error. A fraction of the error is back propagated to the hidden layer and input layer and then the sigmoid function again calculates the actual output. The difference is noted and this process is repeated until the error becomes negligible.

[14] Description of Training Back Propagation Network:

Feedforward

Stage 1. Initialize weights with small, random values

2. While stopping condition is not true – for eachtraining pair (input/output):

•each input unit broadcasts its value to all hidden units

•each hidden unit sums its input signals & applies activation function to compute its output signal

•each hidden unit sends its signal to the outputunits

•each output unit sums its input signals & applies its activation function to compute its output signal

Backpropagation stage

3. Each output computes its error term, its own weight correction term and its bias(threshold) correction term & sends it to layer below

4. Each hidden unit sums its delta inputs from above & multiplies by the derivative of its activation function; it also computes its own weight correction term and its bias correction term

Adjusting the Weights

5. Each output unit updates its weights and bias

6. Each hidden unit updates its weights and bias – Each training cycle is called an epoch. The weights are updated in each cycle – It is not analytically possible to determine where the global minimum is. Eventually the algorithm stops in a low point, which may just be a local minimum.

G. Collaborative learning

Collaborative learning is a process in which many people gather with their data for learning simultaneously. There are two factors impacting the multiparty learning - the number of participating parties and the size of dataset .As the number of participants increases, the operations for each party to share intermediate results and decrypt the final learning result increases.

V. PERFORMANCE EVALUATION

In this section, the several issues related to performance in back propagation algorithm using cloud is discussed. First the accuracy of the output depends on the number of layers in the network. There should be at least three layers in the back propagation configuration network. The more the number of layers, the accuracy of the output increases at the cost of tradeoff with performance. [11] Choosing number of nodes for each layer will depend on problem Neural Network is trying to solve, types of data network is dealing with, quality of data and some other parameters. Number of input and output nodes depends on training set in hand. If there are too many nodes in hidden layer, number of possible computations that algorithm has to deal with increases. Picking just few nodes in hidden layer can prevent the algorithm of its learning ability. Right balance needs to be picked.

Second the security algorithms namely AES and BGN should be applied to only small datasets. So the datasets are made very relevant and apt.

Third the number of iterations in back propagation learning increases the accuracy of the output. The tradeoff with computation time has to be analyzed.

VI. CONCLUSION

The multiple parties participate and share their datasets to make the learning process through back propagation algorithm effective. The Trust Agent distributes the keys in the existing system. The cloud distributes the keys in the proposed system. This is an improvement since the cloud does not know the private data of the owners since they are encrypted first and then uploaded in cloud.

REFERENCES

[1] "The Health Insurance Portability and Accountability Act of Privacy and Security Rules," http://www.hhs.gov/ocr/privacy, 2013.

[2] "National Standards to Protect the Privacy of Personal Health Information," http://www.hhs.gov/ocr/hipaa/finalreg.html, 2013.

[3] M. Abramowitz and I.A. Stegun, Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematics Tables, DoverBooks on Mathematics. Dover, 1964.

[4] A. Bansal, T. Chen, and S. Zhong, "Privacy Preserving Back-Propagation Neural Network Learning over Arbitrarily Partitioned Data," Neural Computing Applications, vol. 20, no. 1, pp. 143150,Feb. 2011.

[5] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF Formulas on Ciphertexts," Proc. Second Int'l Conf. Theory of Cryptography (TCC '05), pp. 325-341, 2005.

[6] T. Chen and S. Zhong, "Privacy-Preserving Backpropagation Neural Network Learning," IEEE Trans. Neural Network, vol. 20, no. 10, pp. 1554-1564, Oct. 2009.

[7] L. Cun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W.Hubbard, and L.D. Jackel, "Handwritten Digit Recognition with a Back-Propagation Network," Proc. Advances in Neural Information Processing Systems, pp. 396-404, 1990.

[8] S.D.C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P.Samarati, "Over-Encryption: Management of Access Control Evolution on Outsourced Data," Proc. 33rd Int'l Conf. Very Large Data Bases (VLDB '07), pp. 123-134, 2007.

[9] T. El Gamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," Proc. Advances in Cryptology (CRYPTO '85), pp. 10-18, 1985.

[10] http://www.m-hikari.com/ces/ces2011/ces1-4-2011/mustafaCES1-4-2011.pdf

[11] http://www.dataminingmasters.com/uploads/studentProjects/NeuralNetworks.pdf

[12] Jiawei Yuan, Student Member, IEEE, and Schucheng Yu, Member, IEEE Privacy Preserving Back Propagation Neural Network learning made Practical with Cloud Computing, "IEEE transactions on parallel and distributed systems, vol no. 1, January 2014.

[13] http://www.cse.unsw.edu.au/~cs9417ml/MLP2/

[14] karmila.staff.gunadarma.ac.id/.../files/.../TayanganBackpropagation.pdf