SCIS 2019 2019 Symposium on
Cryptography and Information Security
Shiga, Japan, Jan. 22 - 25, 2019
The Institute of Electronics,
Information and Communication Engineers

Limitations of Privacy-Preserving for Confidential Data Training by Deep Learning

Harry Chandra Tanuwidjaja* Rakyong Choi* Kwangjo Kim*

Abstract: There is a challenging issue when machine learning algorithm needs to access highly confidential data for the training process. In order to address this problem, several privacy-preserving deep learning, including secure multi-party computing and homomorphic encryption in neural network have been developed. There are also several methods to modify the neural network, so that it can be used in privacy-preserving environment. However, there is trade-off between privacy and performance among various approaches. In this paper, we want to discuss state-of-the-art of privacy-preserving deep learning, evaluate all methods, and compare pros and cons of each approach.

Keywords: privacy-preserving deep learning, homomorphic encryption, trade-off

1 Introduction

The invention of machine learning, i.e., Artificial Intelligence (AI) brings a new era to human life. We can train a machine to do decision making like human being. There are training phase and testing phase in machine learning. In order to get better result, more dataset is required during the training phase. Recently, there is a trend to utilize machine learning in the field of social engineering [1], image recognition [2], and healthcare service [3]. To achieve these applications effectivity, one of the main challenge here is the dataset collection. Since the data will be scattered upon individuals, huge effort to collect them is required.

In general, users tend to reluctantly submit their data to a third party. There is a risk of data leakage, for example when we use cloud computing. Users choose not to store their confidential data in cloud because they worry about that somebody can look at their secret data. In order to convince users for their data security and privacy, there is an approach to use privacy-preserved data to do training process in deep learning. For doing so, the data sent to server is encrypted and during the training process, it will be kept encrypted. The challenge here is to modifying the current deep learning technique, so that it can processes encrypted data. In this paper, we will discuss state-of-the-art of privacy-preserving machine learning approaches, evaluate several known approaches, and compare pros and cons of each approach.

Figure 1 shows the classification of privacy preserving in this paper. The figure presents the applications, metrics, and methods of privacy-preserving approaches. In this paper, we use three metrics to measure the performance of each privacy-preserving deep learning approach, including accuracy, run time, and data transfer. Accuracy means the percentage of correct prediction made by PPDL model. Run time is the time needed by the model to do encryption, sending data from client to server, and doing classification process. Data transfer is the amount of data transferred from client to server. We focus our paper to hybrid Privacy-Preserving Deep Learning (hybrid PPDL) method by combining classical privacy-preserving with various deep learning practices.

The remainder of this paper is organized as follows: Section 2 discusses classical privacy-preserving technology in brief. We examine the original structure of neural network and modification needed for privacy-preserving environment in Section 3. Section 4 presents the analysis of current privacy-preserving deep learning technology. Finally, conclusion and future work are provided in Section 5.

2 Classical Privacy-Preserving Technology

Privacy-preserving technique is classified to a special tool that enables the processing of encrypted data [4]. The importance of privacy-preserving technique is to enable computation on data, without revealing the original content. So, it can ensure the privacy of highly confidential data. Directive 95/46/EC [5] on the protection of individuals with regard to the processing of personal data is a European Union directive that regulates the processing of personal data based on human rights law. The directive states that *"[The data] controller must implement appropriate technical and organizational measures to protect personal data against accidental or unlawful destruction or accidental loss, alteration, unauthorized disclosure or access, in particular where the processing involves the transmission*

* School of Computing, Korea Advanced Institute of Science and Technology (KAIST), 291 Gwahak-ro, Yuseong-gu, Daejeon, 34141, Korea. {*elevantista*, *thepride*, *kkj*}@kaist.ac.kr.

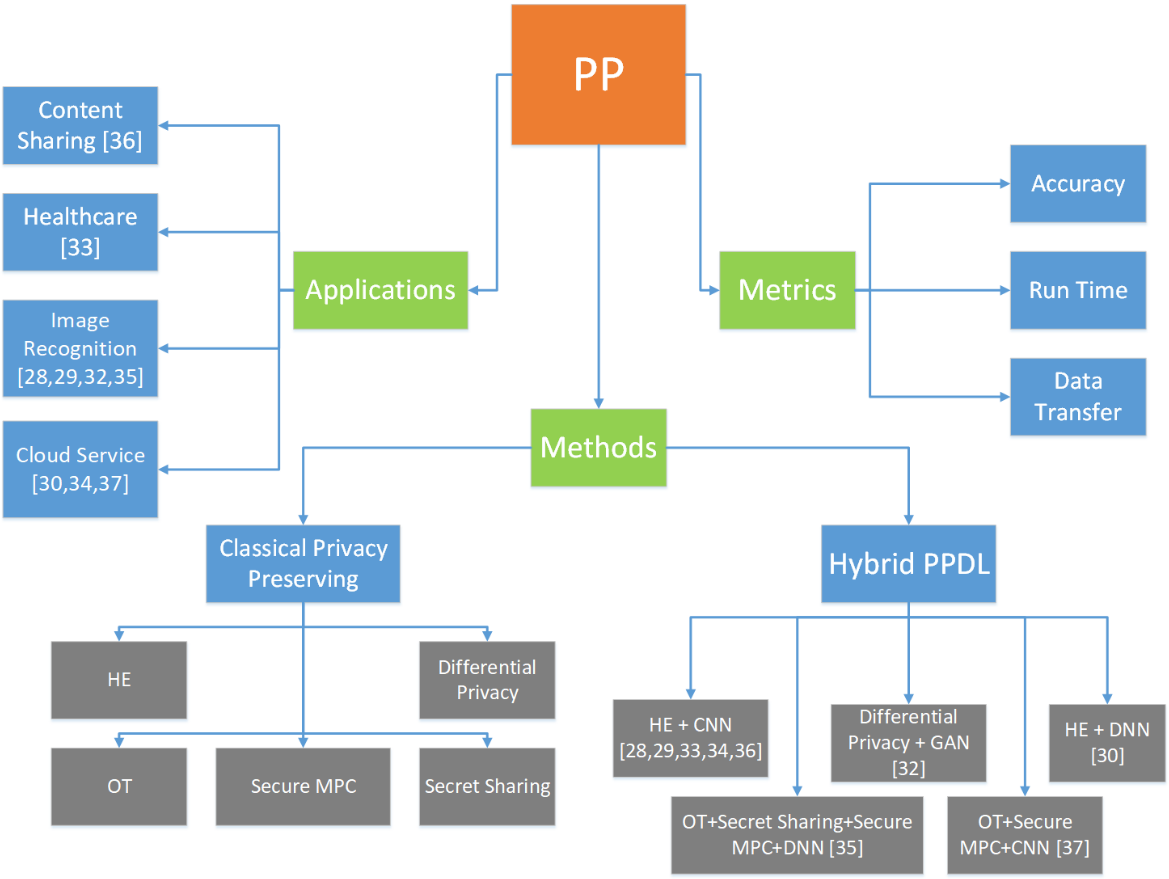


Figure 1: Classification of Privacy-preserving (PP)

of data over a network, and against all other unlawful forms of processing." The goal of privacy-preserving is based on this regulation.

2.1 Homomorphic Encryption

In 1978, Rivest *et al.* [6] questioned whether there exists any encryption scheme that supports the computation on encrypted data without the knowledge of the secret information. For example, the textbook RSA encryption supports multiplication on encrypted data without its private secret key and we call such a system as multiplicative homomorphic encryption (HE). Likewise, we call a system as an additive HE if it supports addition on encrypted data without its secret key.

Fully homomorphic encryption (FHE) means that it supports any computation on encrypted data without the knowledge of the secret key, *i.e.*, for any operation o and two plaintexts m_1, m_2, $Enc(m_1)\ o\ Enc(m_2) = Enc(m_1\ o\ m_2)$. It was remained as an interesting open problem in cryptography for decades till Gentry [7] suggested the first FHE in 2009.

Afterwards, there are a number of research on HE schemes based on lattices with Learning With Errors (LWE) and Ring Learning With Errors (Ring-LWE) problems [8, 9, 10, 11, 12] and schemes over integers with approximate Greatest Common Divisor (GCD) problem [13, 14]. Early work on HE was not practical but for now, there are many cryptographic algorithm tools that supports HE efficiently such as HElib, FHEW, and HEEAN [15, 16, 17].

Homomorphic encryption can be applicable to various areas. As an example, it can improve the security of cloud computing system since it delegates processing of user's data without giving access to the original data. It is also applicable to machine learning methods for encrypted data by outsourcing computation of simple statistics like mean and variance of all original data.

2.2 Secure Multi-party Computation

The concept of secure computation was formally introduced as secure two-party computation in 1986 by Yao [18] with the invention of Garbled Circuit (GC). In GC, all functions are described as a Boolean circuit and an oblivious transfer (OT) protocol is used, to transfer the information obliviously.

Then, Goldreich *et al.* [19] extended the concept to secure multi-party computation in 1987. The purpose of Multi-Party Computation (MPC) is to solve the problem of collaborative computing that keeps privacy of a user in a group of non-trusted users, without using any trusted third party.

Formally, in MPC, for a given number of participants, $p_1, p_2, \cdots, p_n$, each has his private data, $d_1, d_2, \cdots, d_n$, respectively. Then, participants want to compute the value of a public function f on those private data,

$f(d_1, d_2, \cdots, d_n)$ while keeping their own inputs secret.

Compared to HE schemes, in secure MPC, parties jointly compute a function on their inputs using a protocol instead of a single party. During the process, information about parties' secret must not be leaked.

In secure MPC, each party has almost no computational cost with a huge communication cost, while the server has a huge computational cost with almost no communication cost in HE scheme.

To apply secure MPC to deep learning, we must handle the cost of calculating non-linear activation functions like sigmoid or softmax since its cost during training is too large.

2.3 Differential Privacy

Differential privacy was first proposed by Dwork *et al.* in 2005 [20], to treat the problem of privacy-preserving analysis of data.

From the definition in [21], a randomized function $\mathcal{K}$ gives ϵ-differential privacy if for all datasets D_1 and D_2 differing on at most one element, and for all $S \subseteq Range(\mathcal{K})$,

$$\Pr[\mathcal{K}(D_1) \in S] \geq \exp(\epsilon) \times \Pr[\mathcal{K}(D_2) \in S]$$

Differential privacy deals with the case that a trusted data manager wants to release some statistics over his/her data without revealing any information about the data. Thus, an adversary with access to the output of some algorithm learns almost the same information whether user's data is included or not.

Applying differential privacy, there are a number of researches on machine learning algorithms like decision trees, support vector machines, or logistic regressions [22, 23, 24].

3 Deep Learning in Privacy-Preserving Technology

This section describes the original sturucture of deep learning technique and the modification needed for privacy-preserving environment.

3.1 Deep Neural Network (DNN)

3.1.1 Activation Layer

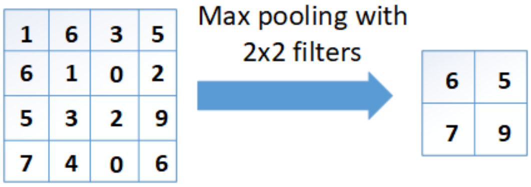
Activation layer, as shown in Figure 2, decides whether the data is activated (value one) or not (value zero). It is located after the convolutional layer. The activation layer is a non-linear function that applies mathematical process on the output of convolutional layer. There are several well-known activation function, such as Rectified Linear Unit (ReLU), sigmoid, and tanh. Because those functions are not linear, the complexity will be really high if we use it to compute the HE encrypted data. So, we need to find a replacement function that only contain multiplication and addition. The replacement function will be discussed later.

x
Activation Function
f(x)

Figure 2: Activation Layer

3.1.2 Pooling Layer

Pooling layer, as shown in Figure 3, is a sampling layer, whose purpose is to reduce the size of data. There are two kinds of pooling technique, max pooling and average pooling. In HE, we cannot use max function, because we are not able to search for the maximum value of encrypted data. As a result, average pooling is the solution to be implemented in HE. Average pooling calculates the sum of values, so there is only addition operation here, which is able to be used over HE encrypted data.



1	6	3	5
6	1	0	2
5	3	2	9
7	4	0	6

6	5
7	9

Figure 3: Pooling Layer

3.1.3 Fully Connected Layer

The illustration of fully connected layer is shown in Figure 4. Each neuron in this layer is connected to neuron in previous layer, so it is called fully connected layer. The connection represents the weight of the feature like a complete graph. The operation in this layer is dot product between the value of output neuron from previous layer and the weight of the neuron. This function is similar to hidden layer in NN. There is only dot product function that consists of multiplication and addition function, so we can use it over HE encrypted data.

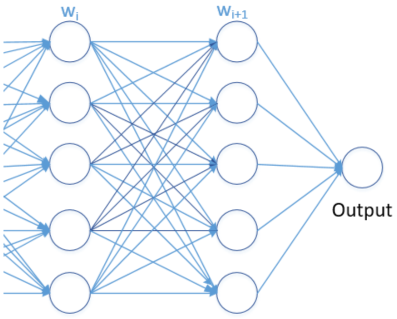


Figure 4: Fully Connected Layer

3.1.4 Dropout Layer

Dropout layer is a layer created to solve over-fitting problem. Sometimes, when we train our machine learning model, the classification result will be too good for some kind of data, which shows bias to the training set. This situation is not good, resulting in huge error during the testing period. Dropout layer will drop random data during training and set it to zero. By doing this iteratively during the training period, we can prevent over-fitting during the training phase.

3.2 Convolutional Neural Network (CNN)

CNN [25] is a class of DNN, which is usually used for image classification. The characteristic of CNN is convolutional layer which purpose is to learn features which are extracted from the dataset. The convolutional layer has $n \times n$ size, which we will do dot product between neighbor values in order to make convolution. As a result, there are only addition and multiplication in convolutional layer. We do not need to modify this layer as it can be used for HE data, a data which is homomorphically encrypted.

3.3 Generative Adversarial Network (GAN)

GAN [26] is a class of DNN, usually used for unsupervised learning. GAN consists of two neural networks that generate candidate model and evaluation model in zero-sum game framework [26]. The generative model will learn samples from dataset until it reaches certain accuracy. On the other hand, the evaluation model discriminates between true data and generated candidate model. GAN does the learning process by modeling the distribution of individual class.

3.4 Modification of Neural Network in Privacy-Preserving Environment

3.4.1 Batch Normalization Layer

Batch Normalization (BN) layer was proposed by Ioffe and Szegedy [27]. The main purpose of BN layer is to fasten the training process by increasing the stability of NN. This layer receives the output from activation layer, then do re-scaling process, resulting in a value between zero and one. BN layer computes the subtraction of each input with the batch mean value, then divides it by the average value of the batch. The BN layer is inserted in every layer of the network to do normalization process for each layer.

3.4.2 Approximation of Activation Function

There have been several researches [4, 28, 29] to do polynomial approximation for activation function. Some well-known methods include numerical analysis, Taylor series, Chebysev polynomial, and polynomial based on the derivative of the activation function. Numerical analysis generates some points from ReLU function, then uses the points as the input of approximation function. However, this method requires high degree for good accuracy, which is not good to be implemented in encrypted data. Taylor series uses polynomials of different degrees to approximate the activation function.

However, it has high degree and the large interval of approximation causes high error rate. Based on Hesamifard *et al.* [28] experiment, the accuracy of Taylor series to approximate ReLU function is around 40%. Chebyshev polynomials approximates activation function based on interval. Their experiment shows that its accuracy is around 70%. Finally, polynomial approach based on the derivative of activation function has the highest accuracy, reaching 99.52%. They calculate the derivative of activation function, approximate the derivative with polynomial, calculate the integral of the polynomial, and finally use it as activation function.

3.4.3 Convolutional Layer with Increased Stride

This architecture is proposed by Liu *et al.* [29] to replace the pooling layer. They leverage convolutional layer with increased stride as a substitution to pooling layer. They use BN layer between the fully connected layer and ReLU. By doing this, the depth of the data stays the same but the dimension is reduced.

4 Analysis of Current Privacy-Preserving Deep Learning Technology

4.1 ML Confidential: Machine Learning on Encrypted Data

Graepel *et al.* [30] proposed ML Confidential, a modified CNN that works on HE scheme. They use polynomial approximation to substitute non-linear activation function. They use cloud service based scenario, and utilize their proposed method to ensure the privacy of data during transfer period between client and server. At first, they do key generation, producing public key and private key for each client. Then, client data is encrypted using homomorphic encryption and transferred to the server. The cloud server will do training process using the encrypted data, and use the training model to do classification on testing dataset.

4.2 On the Protection of Private Information in Machine Learning Systems: Two recent approaches

Abadi *et al.* [31] compares noisy SGD and Private Aggregation of Teacher Ensembles (PATE) [32]. PATE learning process consists of teacher phase and student phase based on differential privacy in GAN. Firstly, during teacher phase, the model is trained using subset of data. Then, the student model will learn from the teacher model. The key of privacy is in teacher model, which is not made public. The advantage of this model is due to the distinguished model, when an adversary is able to get a hold on student model, it will not give them any confidential information. They analyze several aspects, including design, acceptability, work factor, and economy of mechanism. They also show that there is possible failure that reveals some part of training data to the adversary. As a result, notification to the failure is really important, aside from developing cryptography technique for privacy protection.

4.3 Cryptonets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy

Gilad-Bachrach *et al.* [33] proposed Cryptonets, which applies CNN to homomorphically encrypted data. They propose Cryptonets to protect data exchange between user and cloud service. They show that cloud service can apply encrypted prediction based on the encrypted data, then give back the encrypted prediction to user. Later, user can use his own private key to decrypt it, and finally get the prediction result. This scheme can be implemented for hospital service, for example, when a doctor needs to predict the health condition of a patient and take care of an outpatient. Cryptonets uses sigmoid activation function during training phase, but get rid of it during the prediction phase. The weakness of Cryptonets is its performance limitation on the number of non-linear layer. If the number of non-linear layer is big, which we can find at deeper neural network, the error rate will increase and its accuracy drops.

4.4 Privacy-Preserving Classification on Deep Neural Network

Chabanne *et al.* [34] proposed privacy-preserving technique on deep neural network. For the methodology, they combine HE with CNN. Their main idea is to combine Cryptonets [33] with polynominal approximation for activation function and batch normalization layer proposed by Ioffe and Szegedy [27]. They want to improve the performance of Cryptonets, which is only good when the number of non-linear layer in the model is small. The main idea of this paper is changing the structure of regular neural network that consists of convolutional layer, pooling layer, activation layer, and fully connected layer into convolutional layer, pooling layer, batch normalization layer, activation layer, and fully connected layer as it is shown in Figure 5. Max pooling is not a linear function. As a result, in pooling layer they use average pooling, instead of max pooling to provide the homomorphic part with linear function. The batch normalization layer gives contribution to restrict the input of each activation layer, resulting in stable distribution. Polynomial approximation with low degree gives small error, which is very suitable to be used in the model. The training phase is done using the regular activation function, and the testing phase is done using the polynomial approximation, as a substitution to non-linear activation function. Their experiment shows that their model achieves 99.30% accuracy, which is better than Cryptonets (98.95%). The pros of this model is its eligibility to work in neural network with high number of non-linier layers, but still gives accuracy more than 99%, unlike Gilad-Bachrach *et al.* [33] approach that experiences accuracy drop when the number of non-linear layers are increased.

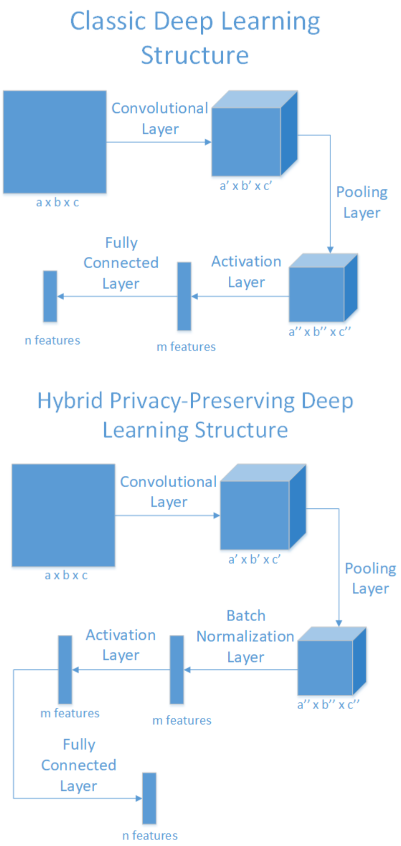


Figure 5: Comparison between regular neural network and privacy-preserving neural network structure

4.5 SecureML: A System for Scalable Privacy-Preserving Machine Learning

Mohassel and Zhang [35] proposed SecureML, a new protocol for privacy-preserving machine learning. They use Oblivious Transfer (OT), Yao's GC, and Secret Sharing. OT is a security protocol proposed by Rabin [36], in which the sender of message remains oblivious whether the receiver has got the message or not. Secret sharing becomes one of basic cryptographic tools to distribute a secret between parties since Shamir [37] proposed the first secret sharing scheme in 1979. Shamir's secret sharing scheme is a kind of threshold secret sharing schemes since it requires the minimum number of secret shares to recover the secret. For deep learning part, they leverage linear regression and logistic regression in DNN environment. They propose addition and multiplication algorithm for secretly shared values in linear regression. The authors leverage Stochastic Gradient Descent (SGD) method in order to calculate the optimum value of regression. The weakness of this scheme is that they can only implement a simple neural network, without any convolutional layer, so the accuracy as summarized in Table 1, is quiet low.

4.6 CryptoDL: Deep Neural Networks Over Encrypted Data

Hesamifard *et al.* [28] proposed CryptoDL, a modified CNN for encrypted data. They change the activa-

Table 1: Comparison per each method and performance for each privacy-preserving deep learning

Year	Publication	PP	DL	Scenario	Accuracy (%)	Run Time (s)	Data Transfer (MB)
2012	GLN12 [30]	HE	DNN	Cloud Service	95.00	255	N/A
2016	PAE16 [32]	Differential Privacy	GAN	Image Recognition	98.10	N/A	N/A
2016	BDLL16 [33]	HE	CNN	Healthcare	98.95	697	595.5
2017	CWMM17 [34]	HE	CNN	Cloud Service	99.30	N/A	N/A
2017	MZ17 [35]	OT, Yao's GC, Secret Sharing	DNN	Image Recognition	93.40	N/A	N/A
2017	HTG17 [28]	HE	CNN	Image Recognition	99.52	320	336.7
2018	LPWCT18 [29]	HE	CNN	Image Recognition	98.97	477	361.6
2018	XHLQ18 [38]	HE	CNN	Content Sharing	99.73	N/A	N/A
2018	RRK18 [39]	OT, Yao's GC	CNN	Cloud Service	98.95	10649	722000

tion function part of CNN with low degree polynomial. This paper shows that the polynomial approximation is indispensable for neural network in homomorphic encryption environment. They try to approximate three kinds of activation function: ReLU, sigmoid, and tanh. The approximation technique is based on the derivative of activation function. Firstly, during training phase, CNN with polynomial approximation is used. Then, the model produced during the training phase is used to do classification over encrypted data. The authors apply their method to MNIST dataset [40], and achieve 99.52% accuracy. The weakness of this scheme is not covering privacy-preserving training in deep neural network. They use the privacy-preserving for classification process only. The pros of this work is it can classify many instances (8192 or larger) for each prediction round, unlike *Rouhani et al.* [39] that classifies one instance per round. So we can say that CryptoDL works more effective compared to DeepSecure [39].

4.7 Privacy-Preserving All Convolutional Net Based on Homomorphic Encryption

Liu *et al.* [29] proposed privacy-preserving technique on convolutional network by using HE. They use MNIST dataset [40] that contains handwritten number. They encrypt the data using HE, then use the encrypted data to train Convolutional Neural Network (CNN). Later, they do classification and testing process using the model from CNN. Their idea is adding batch normalization layer before each activation layer and approximate activation layer using Gaussian distribution and Taylor series. They also change the non-linear pooling layer with convolutional layer with increased stride. By doing this, they have successfully modified CNN to be compatible with HE, and achieve 98.97% accuracy during the testing phase. We can see that the main difference between regular CNN and modified CNN in privacy-preserving technology is the addition of batch normalization layer and the change of non-linear function in activation layer and pooling layer into linear function.

4.8 Distributed Large Scale Privacy-Preserving Deep Mining

Xue *et al.* [38] proposed privacy-preserving deep learning using multi-key FHE. They do some modification to conventional CNN structure, such as changing max pooling into average pooling, adding batch normalization layer before each activation function layer, and replacing ReLU activation function with low degree approximation polynomial. Their method is beneficial for classifying large scale distributed data, for example, in order to predict future road condition, we need to train neural network model from traffic information data which are collected from many cars. The security and privacy issue during data collection and training process can be solved using their approach.

4.9 Deepsecure: Scalable Provably-Secure Deep Learning

Rouhani *et al.* [39] proposed DeepSecure, a framework that enables the use of deep learning in privacy-preserving environment. The authors use OT and Yao's GC protocol [18] with CNN to do the learning process. DeepSecure enables a collaboration between client and server to do learning process on cloud server using data from client. They do security proof of their system by using Honest-but-Curious (HbC) [11] adversary model. It has been successfully shown that the GC protocol keep the client data private during the data transfer period. The cons of this method is its limitation of number of instance processed each round. They are only able to classify one instance during each prediction round.

4.10 Analysis of the Previous Works

The comparison per each method and performance for each privacy-preserving deep learning approach can be seen in Table 1. We can see that most of the approaches leverage homomorphic encryption as their cryptography technique and Convolutional Neural Network as their Deep Learning technique. The best accuracy is given by Xue *et al.* [38] with 99.73% accuracy. Their high performance is caused by combining ideas from previous works, such as substituting ReLU function with low degree polynomial, using batch normalization layer, and multi-key FHE to support the large scale multi-user environment. From our analysis above, we believe that main challenge in privacy-preserving machine learning technique regards to the trade-off between accuracy and complexity. If we use high degree polynomial approximation for activation function, the accuracy will become better, but in cost for high complexity. On the other hand, low degree polynomial approximation for activation function gives low complexity with worse accuracy compared to high degree polynomial. Choosing correct approximation method for each privacy-preserving scenario is the main challenge here.

5 Conclusion and Future Work

In this paper, we have discussed state of the art of privacy-preserving deep learning. We analyze the original structure of neural network and the modification needed to use it in privacy-preserving environment. We also address the trade-off between accuracy and complexity during the substitution process of non-linear activation function as the main challenge. An open problem regarding privacy-preserving machine learning technique is to reduce computational burden. How to divide the burden between a client and a server optimally, to get the best performance is a big challenge that needs to be addressed in the future.

Acknowledgement

This work was partly supported by Indonesia Endowment Fund for Education (LPDP) and Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00555, Towards Provable-secure Multi-party Authenticated Key Exchange Protocol based on Lattices in a Quantum World).

References

[1] D. Lazer, A. S. Pentland, L. Adamic, S. Aral, and A. L. Barabasi, "Life in the network: the coming age of computational social science," *Science*, vol. 323, p. 721, 2009.

[2] N. M. Nasrabadi, "Pattern recognition and machine learning," *Journal of electronic imaging*, vol. 16, 2007.

[3] M. Chen, Y. Hao, K. Hwang, and L. Wang, "Disease prediction by machine learning over big data from healthcare communities," *IEEE Access*, vol. 5, pp. 8869–8879, 2017.

[4] D. Zhang, X. Chen, D. Wang, and J. Shi, "A survey on collaborative deep learning and privacy-preserving," *IEEE Third International Conference on Data Science in Cyberspace*, pp. 652–658, 2018.

[5] M. Meints and J. Moller, "Privacy preserving data mining: a process centric view from a European perspective," 2004.

[6] R. L. Rivest, L. Adleman, and M. L. Dertouzos, "On data banks and privacy homomorphisms," in *Foundations of secure computation 4.11*, 1978, pp. 169–180.

[7] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Annual ACM on Symposium on Theory of Computing.* ACM, 2009, pp. 169–178.

[8] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 831–871, 2014.

[9] ——, "Fully homomorphic encryption from ring-LWE and security for key dependent messages," in *Advances in Cryptology–CRYPTO 2011.* Springer, 2011, pp. 505–524.

[10] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in *Advances in Cryptology–CRYPTO 2013.* Springer, 2013, pp. 75–92.

[11] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(leveled) fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory (TOCT)*, vol. 6, no. 3, p. 13, 2014.

[12] M. Clear and C. McGoldrick, "Multi-identity and multi-key leveled fhe from learning with errors," in *Annual Cryptology Conference.* Springer, 2015, pp. 630–656.

[13] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully homomorphic encryption over the integers," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques.* Springer, 2010, pp. 24–43.

[14] J. H. Cheon, J.-S. Coron, J. Kim, M. S. Lee, T. Lepoint, M. Tibouchi, and A. Yun, "Batch fully homomorphic encryption over the integers," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques.* Springer, 2013, pp. 315–335.

[15] S. Halevi and V. Shoup, "Algorithms in helib," in *International Cryptology Conference.* Springer, 2014, pp. 554–571.

[16] L. Ducas and D. Micciancio, "Fhew: bootstrapping homomorphic encryption in less than a second," in *Annual International Conference on the*

Theory and Applications of Cryptographic Techniques. Springer, 2015, pp. 617–640.

[17] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2017, pp. 409–437.

[18] A. C.-C. Yao, "How to generate and exchange secrets," in *Foundations of Computer Science, 1986., 27th Annual Symposium on*. IEEE, 1986, pp. 162–167.

[19] O. Goldreich, S. Micali, and A. Wigderson, "How to play any mental game," in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*. ACM, 1987, pp. 218–229.

[20] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography conference*. Springer, 2006, pp. 265–284.

[21] C. Dwork, "Differential privacy," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2006, pp. 1–12.

[22] K. Chaudhuri, C. Monteleoni, and A. D. Sarwate, "Differentially private empirical risk minimization," *Journal of Machine Learning Research*, vol. 12, no. Mar, pp. 1069–1109, 2011.

[23] D. Kifer, A. Smith, and A. Thakurta, "Private convex empirical risk minimization and high-dimensional regression," in *Conference on Learning Theory*, 2012, pp. 25–1.

[24] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright, "A practical differentially private random decision tree classifier," in *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on*. IEEE, 2009, pp. 114–121.

[25] e. a. LeCun, Yann, "Object recognition with gradient-based learning," *Shape, contour and grouping in computer vision*, pp. 319–345, 1999.

[26] I. e. a. Goodfellow, "Generative adversarial nets." in advances in neural information processing systems," *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint*, vol. 1502.03167, 2015.

[28] E. Hesamifard, H. Takabi, and M. Ghasemi, "Cryptodl: Deep neural networks over encrypted data," *arXiv preprint*, vol. 1711.05189, 2017.

[29] W. Liu, F. Pan, X. A. Wang, Y. Cao, and D. Tang, "Privacy-preserving all convolutional net based on homomorphic encryption," *International Conference on Network-Based Information Systems*, pp. 752–762, 2018.

[30] T. Graepel, K. Lauter, and M. Naehrig, "Ml confidential: Machine learning on encrypted data," *International Conference on Information Security and Cryptology*, pp. 1–21, 2012.

[31] M. Abadi, U. Erlingsson, and I. Goodfellow, "On the protection of private information in machine learning systems: Two recent approches," *Computer Security Foundations Symposium*, pp. 1–6, 2017.

[32] N. Papernot, M. Abadi, and U. Erlingsson, "Semi-supervised knowledge transfer for deep learning from private training data," *arXiv preprint*, vol. 1610.05755, 2016.

[33] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," *International Conference on Machine Learning*, pp. 201–210, 2016.

[34] H. Chabanne, A. de Wargny, J. Milgram, C. Morel, and E. Prouff, "Privacy-preserving classification on deep neural network," *IACR Cryptology ePrint Archive*, p. 35, 2017.

[35] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," pp. 19–38, 2017.

[36] M. O. Rabin, "How to exchange secrets with oblivious transfer," *IACR Cryptology ePrint Archive*, p. 187, 2005.

[37] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[38] H. Xue, Z. Huang, H. Lian, W. Qiu, J. Guo, S. Wang, and Z. Gong, "Distributed large scale privacy-preserving deep mining," *IEEE Third International Conference on Data Science in Cyberspace*, pp. 418–422, 2018.

[39] B. Rouhani, M. Riazi, and F. Koushanfar, "Deepsecure: Scalable provably-secure deep learning," *55th ACM/ESDA/IEEE Design Automation Conference*, pp. 1–6, 2018.

[40] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, pp. 141–142, 2012.