A Survey on Deep Learning Techniques for Privacy-Preserving

Kwangjo Kim, Harry Chandra Tanuwidjaja, and Rakyong Choi
Korea Advanced Institute of Science and Technology (KAIST)
August 18, 2019

CONTENTS
1. Introduction
2. Classical Privacy-Preserving Technologies
3. Deep Learning in Privacy-Preserving Technologies
4. X-based Hybrid Privacy-Preserving Deep Learning
5. Comparison
6. Conclusion and Future Work

History of Deep Learning: Ideas and Milestone
- 1943: Neural networks
- 1957: Perceptron
- 1974-86: Backpropagation, RNN, RNN
- 1889-98: CNN, MNIST, Bidirectional RNN
- 2006: Deep Learning
- 2009: ImageNet
- 2012: AlexNet, Dropout
- 2014: GAN (Generative Adversarial Networks)
- 2016: DeepFace
- 2016: AlphaGo
- 2018: AlphaZero, Capsule Networks
- 2018: BERT (Bidirectional Encoder Representations from Transformers) by Google

Why we need Privacy-Preserving Deep Learning?
- Advances of machine learning
- Users (Data Owner) submit data to the trustful cloud server who want to get useful statistics of users
- Data privacy during training
- Solution?
  - Privacy Preserving Deep Learning (PPDL)

https://deeplearning.mit.edu
Our Classification

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>PP</td>
<td>Privacy Preserving</td>
</tr>
<tr>
<td>DL</td>
<td>Deep Learning</td>
</tr>
<tr>
<td>HE</td>
<td>Homomorphic Encryption</td>
</tr>
<tr>
<td>OT</td>
<td>Oblivious Transfer</td>
</tr>
<tr>
<td>MPC</td>
<td>Multi Party Computing</td>
</tr>
<tr>
<td>CNN</td>
<td>Convolutional Neural Network</td>
</tr>
<tr>
<td>DNN</td>
<td>Deep Neural Network</td>
</tr>
<tr>
<td>BNN</td>
<td>Binary Neural Network</td>
</tr>
</tbody>
</table>

Classical Privacy-Preserving Technology

- Homomorphic Encryption
  - Support operations on encrypted data without private key
  - Not directly applicable to DL
- Secure Multi-party Computation
  - Joint computation of f(), keeping each input to be secret
- Differential Privacy
  - Keeping privacy before and after PP
  - Release statistics without revealing data

Deep Learning in Privacy-Preserving Technology

- Deep Neural Network (DNN)

- Convolutional Neural Network (CNN)
Deep Learning Layers (1/5)

- **Convolutional Layer**
  - Apply a convolution operation to the input, passing the result to the next layer.
  - **Dot product** operation
  - Can be used directly in HE

Deep Learning Layers (2/5)

- **Activation Layer**
  - **Non-linear** function that applies mathematical process on the output of convolutional layer.
  - Activation function: ReLU, Sigmoid, Tanh
  - Non-linear -> high complexity

Deep Learning Layers (3/5)

- **Pooling Layer**
  - A sampling layer, whose purpose is to reduce the size of data
  - Cannot use max pooling in HE
  - Solution? **Average pooling**

Deep Learning Layers (4/5)

- **Fully Connected Layer**
  - Each neuron in this layer is connected to neuron in previous layer
  - The connection represents the weight of the feature like a complete graph
  - **Dot product** function
  - Can be used directly in HE
Deep Learning Layers (5/5)

- **Dropout Layer**
  - Reduce overfitting, act as regularizer
  - Not using all neurons
  - Drops some neurons randomly

X-based Hybrid PPDL

- HE-based Hybrid PPDL
- Secure MPC-based Hybrid PPDL
- Differential Privacy-based Hybrid PPDL

HE-based Hybrid PPDL

- **ML Confidential**: Machine Learning on Encrypted Data
  - Polynomial approximation as activation function
  - Cloud based scenario
  - Homomorphic encryption
  - Data is transferred to server
  - Cloud server does training process;

**Crypto-PPDL 2019 HE-based Hybrid PPDL(2/10)**

- **Cryptonets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy**
  - Protect data exchange in cloud service
  - Apply CNN to homomorphically encrypted data
  - **Weakness:** error rate increase and accuracy drops
    - When?
    - If the number of non-linear layer is big

---

**International Conference on Machine Learning, pp. 201-210, 2016.**

---

**Crypto-PPDL 2019 HE-based Hybrid PPDL(3/10)**

- **Privacy-Preserving on Deep Neural Network**
  - Cloud service environment
  - Combining HE with CNN
  - Solve Cryptonets problem
  - Polynomial approximation
  - Batch normalization layer

---

**H. Chabanne, A. de Wargny, J. Milgram, C. Morel, and E. Prou, “Privacy-preserving classification on deep neural network,”**
**IACR Cryptology ePrint Archive, p. 35, 2017.**

---

**Crypto-PPDL 2019 HE-based Hybrid PPDL(4/10)**

- **CryptoDL: Deep Neural Networks Over Encrypted Data**
  - Modified CNN for encrypted data with HE
  - Approximation technique:
    - Taylor series (Acc 40%)
    - Chebyshev polynomial (Acc 70%)
    - Derivative of activation function (Acc 99.52%)

---


---

**Crypto-PPDL 2019 HE-based Hybrid PPDL(5/10)**

- **Privacy-Preserving All Convolutional Net Based on Homomorphic Encryption**
  - PP technique on CNN by using HE
  - Adding batch normalization layer
  - Polynomial approximation
  - Convolution layer with increased stride

---

**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.**
HE-based Hybrid PPDL(6/10)

- Distributed Privacy-Preserving Multi-Key Fully Homomorphic Encryption
  - Substituting ReLU function with low degree polynomial
  - Using batch normalization layer
  - Max pooling -> average pooling
  - Beneficial for classifying large scale distributed data


HE-based Hybrid PPDL(7/10)

- Gazelle: A Low Latency Framework for Secure Neural Network Inference
  - Able to switch protocol between HE and GC in PaaS scenario.
  - Structure: two convolutional layers, two ReLU layers, one pooling layer, and one fully connected layer.
  - Hide the weight, bias, and stride size in the convolutional layer.
  - Limit the number of classification queries from client to prevent linkage attack.


HE-based Hybrid PPDL(8/10)

- Tapas
  - Accelerate parallel computation using encrypted data in PaaS environment.
  - Current problem: large amount of processing time needed.
  - Main contribution:
    - New algorithm to speed up binary computations in Binary Neural Network (BNN).
  - Their technique can be parallelized by evaluating gates at the same level for three representations at the same time -> time improved drastically.


HE-based Hybrid PPDL(9/10)

- FHE DiNN
  - Reduce complexity problem in HE+NN
  - Deeper network, more complexity
  - Use bootstrapping -> linear complexity of NN
  - How to do it?
    - Discretize the weight, bias value, and the domain of activation function.
    - Using sign activation function to limit the growth of signal in the range of [-1,1]

E2DM

- PPDL framework that performs matrix operations on HE system
- Encrypts a matrix homomorphically, then do arithmetic operations on it.
- Reduce complexity of matrix multiplication
  - $O(d)$ complexity for dot product between two $d \times d$ matrices
  - instead of $O(d^2)$ complexity.
- Leverage CNN with one convolutional layer, two fully connected layers, and a square activation function.


---

Metrics for Comparison

<table>
<thead>
<tr>
<th>PDNL Metrics</th>
<th>Accuracy</th>
<th>Run Time</th>
<th>Data Transfer</th>
<th>PoC (Privacy of Client)</th>
<th>PoM (Privacy of Model)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>% of correct prediction made by used PPDL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run time</td>
<td>the total time of encryption, sending data from client to server, and classification process.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data transfer</td>
<td>the amount of data transferred from client to server.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PoC</td>
<td>neither the server or any other party knows about client data.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PoM</td>
<td>neither the client or any other party knows about the classification model used on server.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

Comparison of HE-based PPDL

<table>
<thead>
<tr>
<th>Cloud Service</th>
<th>Proposed Technique</th>
<th>HE Techniques</th>
<th>Accuracy (%)</th>
<th>Run Time</th>
<th>Data Transfer</th>
<th>PoC</th>
<th>PoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>HE (BFV)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CNN</td>
<td>HE (BMP)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CNN</td>
<td>HE (BFV + SHE)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>CNN</td>
<td>HE (BMP + SHE)</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Secure MPC-based Hybrid PPDL
SecureML: A System for Scalable Privacy-Preserving Machine Learning

- Based on OT, Yao’s GC, and secret sharing
- The sender of message remains oblivious: whether the receiver has got the message or not
- Linear regression and logistic regression
- Optimum value of regression?
  - Stochastic Gradient Descent (SGD)

Deepsecure: Scalable Provably-Secure Deep Learning

- Use OT and Yao’s GC protocol with CNN
- Collaboration between client and server
- Weakness: limited number of instances processed
- Only able to classify one instance during each round

MiniONN

- PP framework that transforms a NN into an oblivious NN.
- Two kinds of transformations:
  - Piecewise linear activation function
  - Oblivious transformation for smooth activation function
- Supports all activation functions that have:
  - Monotonic range
  - Piecewise polynomial, or
  - Can be approximated into polynomial function.

ABY3

- PPDL framework based on three-party computation
- Can switch between arithmetic, binary, and Yao’s 3PC
- Use binary sharing on three-party Garbled Circuit
- Arithmetic sharing when training linear regression model
- Outperform MiniONN by four orders of magnitude faster
**Comparison of MPC-based PPDL**

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Proposed Scheme</th>
<th>DL Technique</th>
<th>Accuracy (%)</th>
<th>Time (s)</th>
<th>Data Transfer (MB)</th>
<th>PoC</th>
<th>PoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chao</td>
<td>Stochastic</td>
<td>CNN</td>
<td>Good</td>
<td>30.54</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Beaten</td>
<td>Bayesian</td>
<td>Good</td>
<td>30.54</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td></td>
<td>E2DM</td>
<td>NN</td>
<td>Good</td>
<td>30.54</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td></td>
<td>E2DM</td>
<td>AN</td>
<td>Good</td>
<td>30.54</td>
<td>Yes</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>

**Differential Privacy-based PPDL**

- Private Aggregation of Teacher Ensembles (PATE)
  - Teacher phase and student phase
  - Possible failure that reveals some part of training data

- E2DM gives the best performance:
  - High accuracy
  - Fast run time
  - Small data transfer

**Comparison-All**

- Table showing performance metrics for different schemes.
Conclusion and Future Work

- Discussed state of the art of privacy-preserving deep learning
- Layers modified in PPDL:
  - pooling layer, activation layer, and batch normalization layer
- Future Work:
  - Achieving more than 99% accuracy with good PoC and PoM
  - Lots of Challenges still remain

---

Q&A

- Newly published paper that improve E2DM
- Substitute HE with functional Encryption (FE)
- Compute in:
  - HE: Encrypted form
  - CryptoNN: Plaintext
- Performance: 95.50% accuracy

---

CryptoNN: Training Neural Networks over Encrypted Data

- There are 2 main parts:
  - FE part
    - Authority manages key generation
    - FHE
    - Feedforward network
    - Redeployment to minimize cost
- Newly published paper that improve E2DM
- Substitute HE with functional Encryption (FE)
- Compute in:
  - HE: Encrypted form
  - CryptoNN: Plaintext
- Performance: 95.50% accuracy