SHE: A Fast and Accurate Deep Neural Network for Encrypted Data

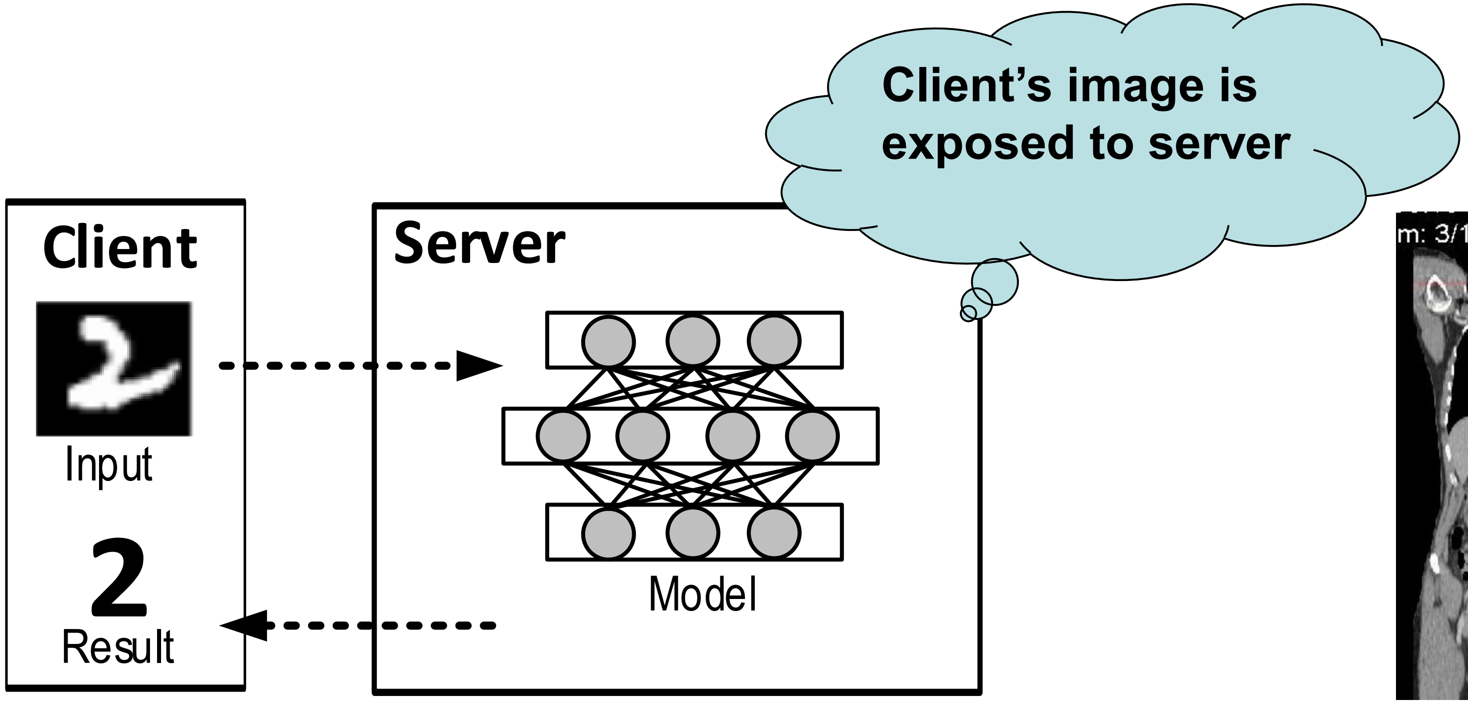
Qian Lou, Lei Jiang

Indiana University Bloomington

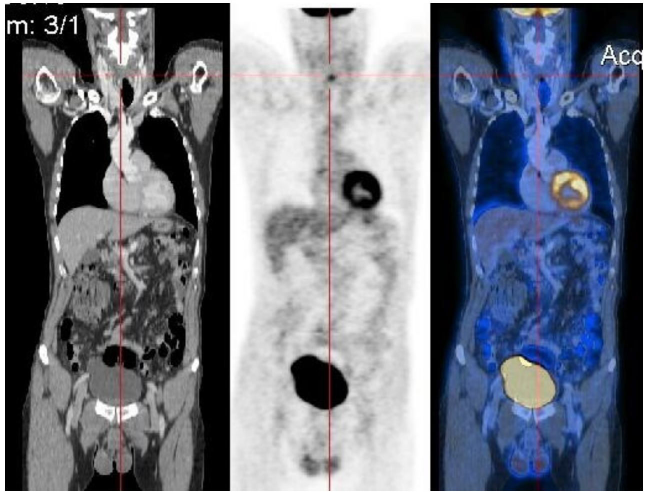
Outline

1. MLaaS and Data Privacy
2. Secure MLaaS
3. Our work
4. Conclusion

Machine Learning as a Service(MLaaS)



MLaaS



Private data

MLaaS needs to protect client's data privacy

Homomorphic Encryption

- Homomorphism
  - Addictive homomorphism: F(a+b)=F(a)+F(b)
  - Multiplicative homomorphism: F(a*b)=F(a)*F(b)
- Homomorphic Encryption: let **F()** is an encryption **Enc()**
  - Addition on ciphertext:
    Enc(a+b)=Enc(a)+Enc(b)
  - Multiplication on ciphertext:

    Enc(a*b)=Enc(a)*Enc(b)

Secure addition example

Task : Server helps client compute (15=5+10) but server dose not know the number 5 and 10.

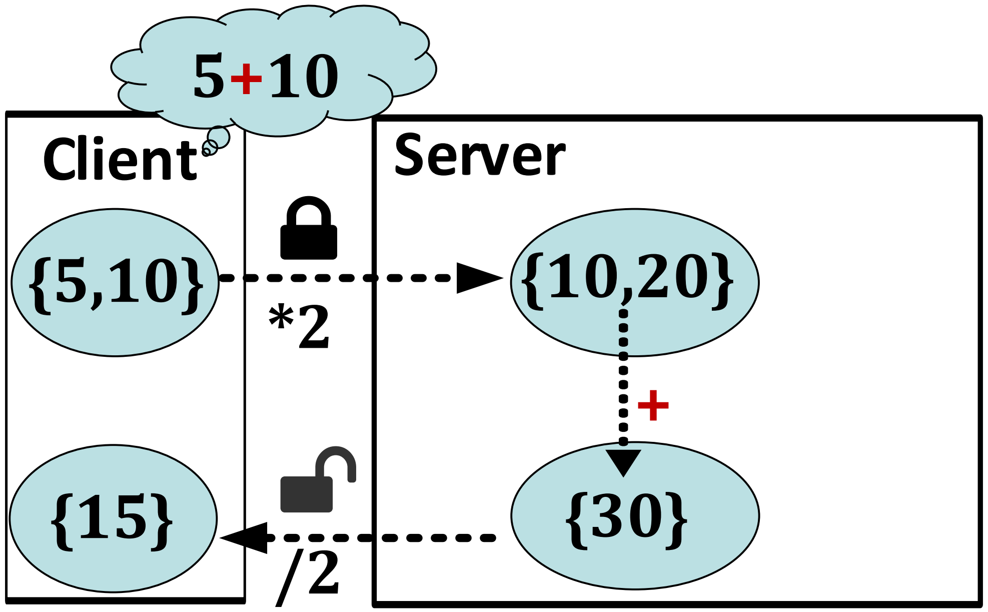
Solution: addition on ciphertext:

Let encryption function Enc(x)=2*x; decryption function Dec(x)=x/2;

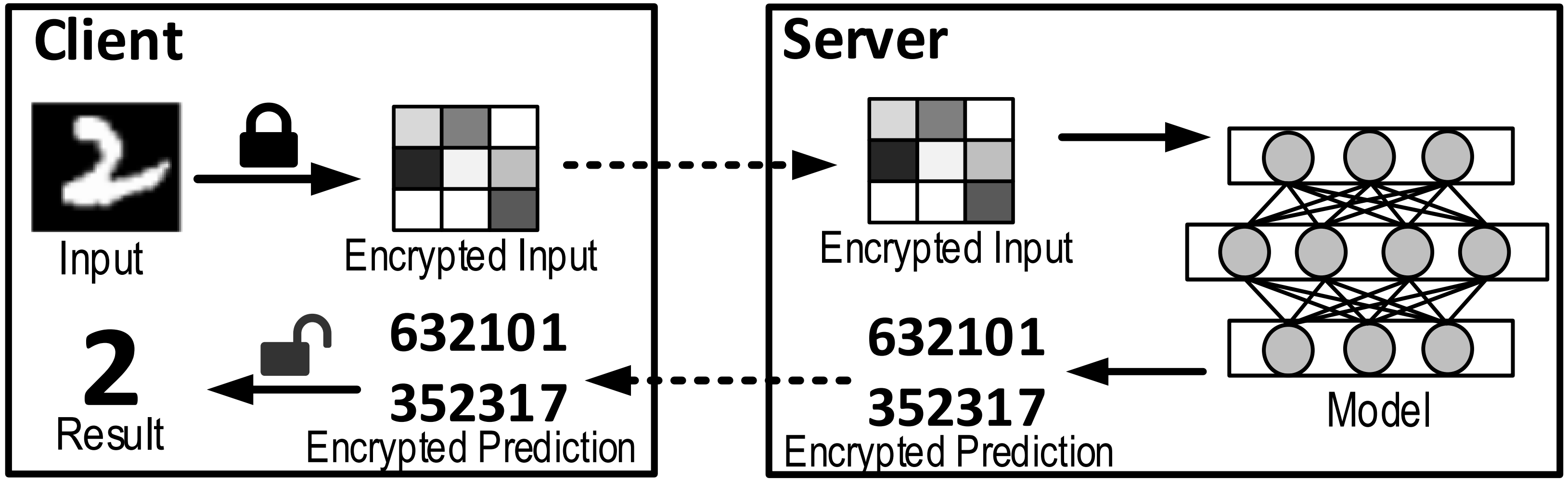
1. Encryption: Enc(5)=10; Enc(10)=20.

2. Addition:
   Enc(5+10)=Enc(5)+Enc(10) =30;

3. Decryption:
   Dec(Enc(5+10))=30/2=15



HE enables secure MLaaS



Secure MLaaS

- 1. Encrypts input
- 2. Uploads encrypted input
- 3. Performs inference on encrypted data
- 4. Downloads encrypted prediction
- 5. Decrypts encrypted prediction

Problems

- HE only supports linear operations
  - **Addictive homomorphism** ($F(a+b)=F(a)+F(b)$)
  - **Multiplicative homomorphism** ($F(a*b)=F(a)*F(b)$)

- Deep Neural Network (DNN)
  - **Linear operations** (Convolution Layer, Dense Layer)
  - **Non-linear Operations** (Activation functions, Max Pooling)

How to process non-linear operations?

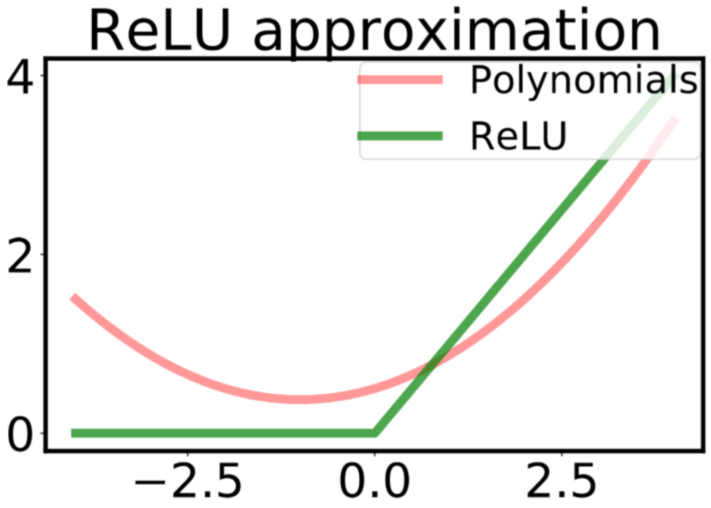
- Previous works: Polynomial approximation

  - **ReLU:** $Y = \max(x,0) \approx x^2 \approx 0.125x^2+0.25x+0.5$

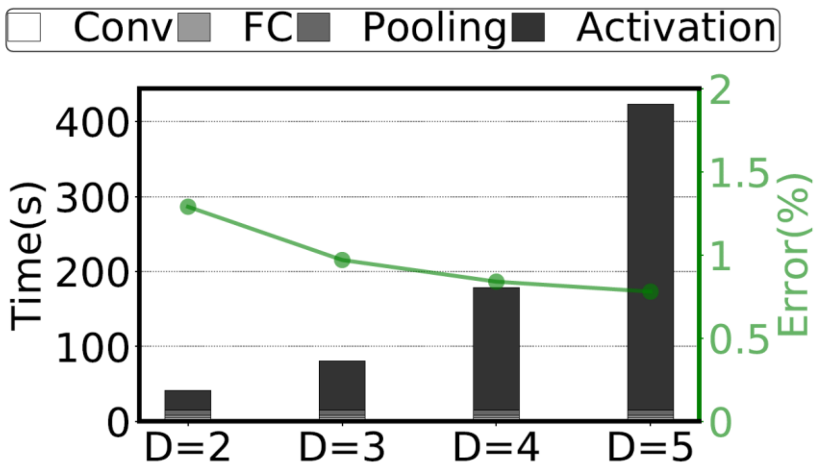
Outline

1. MLaaS and Data Privacy
2. Secure MLaaS
3. Our work
4. Conclusion

Motivation



1. Inaccurate approximation



2. Various-degrees approximation

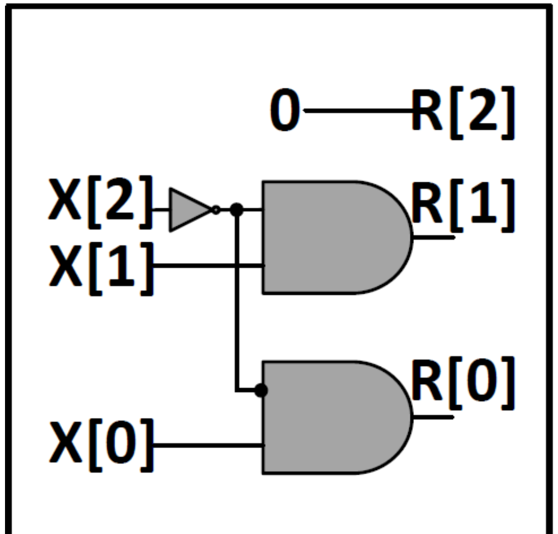
Executive summary

- **Need**: Fast and accurate deep Learning over encrypted data
- **Opportunities** to improve privacy-preserving deep learning by the co-design of Homomorphic Encryption scheme and neural network optimization:
  - Binary bits-operations-friendly TFHE encryption scheme
  - Shift-Accumulation based quantization for neural network
- **Problem**: Previous works stacked **multiple** & **inaccurate** ReLU activation and max pooling layers (Polynomials approximation):

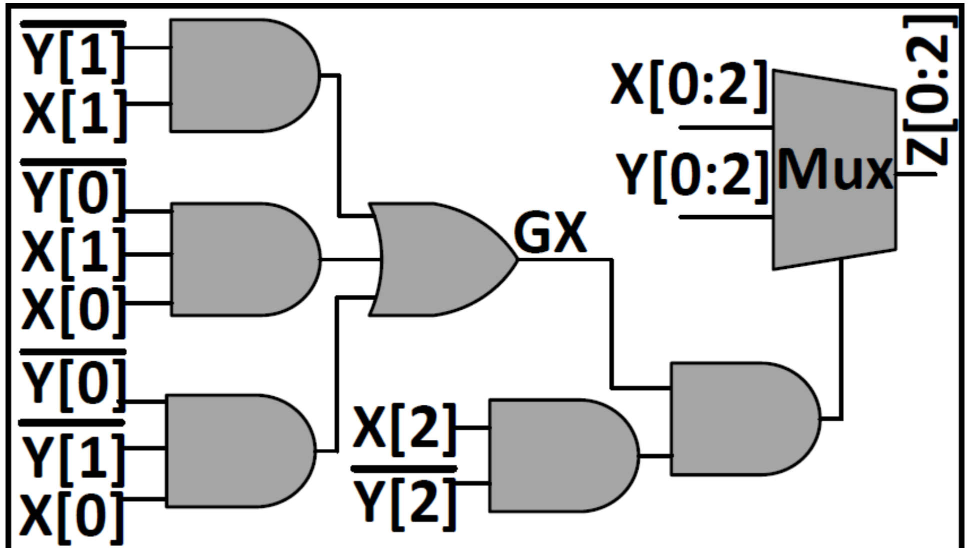
  Accuracy ↓ & overhead ↑ & shallow networks topology
- **Key Idea**: Directly implementing ReLU and max using TFHE [1]; Using cheap Shift-Accumulation to support deeper neural networks other than acceleration.
- **SHE**: Accuracy-lossless CNN, performance ↑76.12%, the first to support modern deep learning like AlexNet on ImageNet.

Our work SHE

- **SHE**
  - Supports **accuracy-lossless** ReLU and Max → **Accurate**
  - TFHE scheme



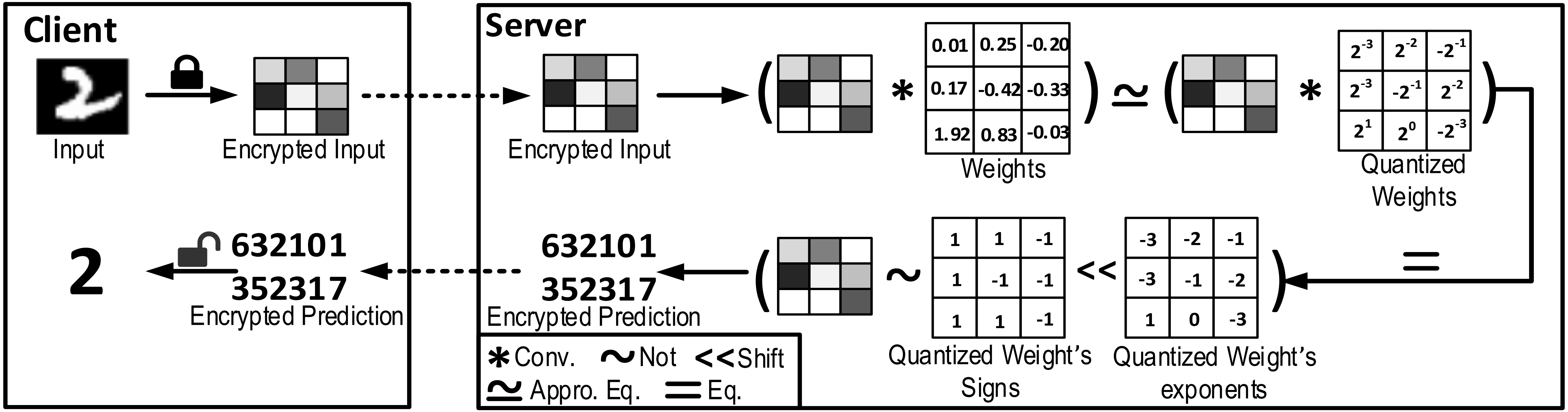
3-bit ReLU

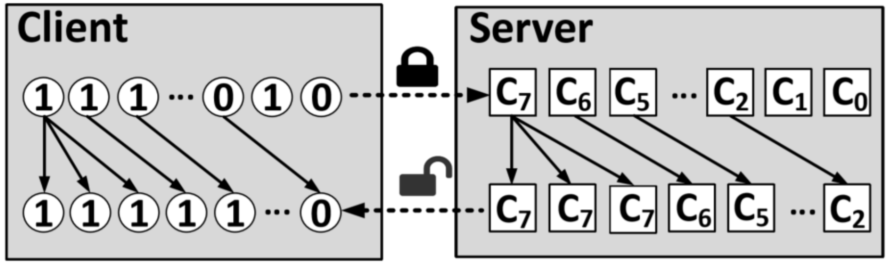


3-bit Max

Our work SHE

- **SHE**
  - Logarithmic Quantization: Convolution to Shift-Accumulation → **Fast**





Very Cheap shift operations

Thank you!