Training a Logistic Regression Model on Encrypted Data

Lab Machine Learning on Encrypted Data WS2020

Gerd Mund, Van Thong Nguyen, Yat Wai Wong

26.03.2021

Contents

Overview of Homomorphic Encryption

- Computation on encrypted data without the need of decryption or decryption key
- i.e. instead of decrypting the data then compute the function $f(m_1, m_2)$ on plaintexts, HE allows us to compute on encrypted data: $f_{HE}(Enc(m_1), Enc(m_2))$
- e.g. Homomorphic addition $Enc(m_1) +_{HE} Enc(m_2)$ - sum of ciphertexts correctly decrypted to $(m_1 + m_2)$.

Partially HE, somewhat HE

- Supports either addition or multiplication
- Paillier supports homomorphic addition
- RSA, ElGamel support homomorphic multiplication
- Partially homomorphic - limited applications.
- Somewhat HE: both homomorphic addition and multiplication, until ciphertexts cannot correctly be decrypted
- Fully HE: with bootstrapping, no limit of number of addition/multiplication

FHE, Bootstrapping

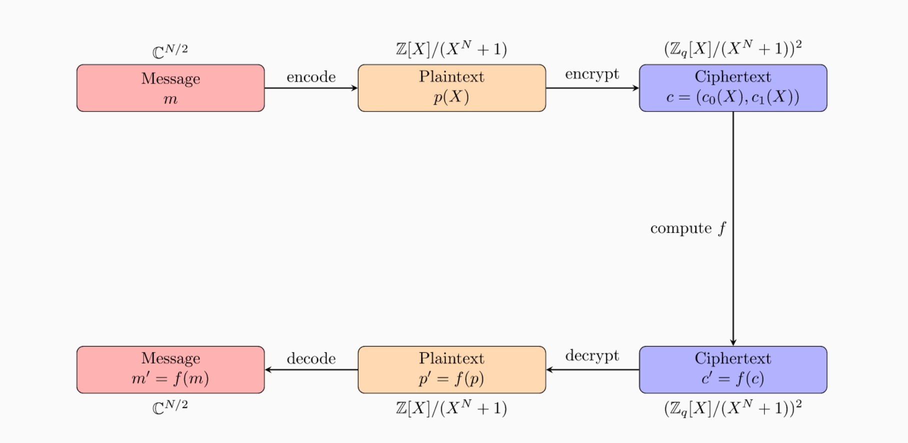
- Bootstrapping proposed by Gentry [1] to enable unbounded homomorphic computation
- Refresh ciphertexts by homomorphic decryption, results in lower error rate
- Encrypt the secret key, then homomorphically decrypt ciphertext
- Suppose $\mathrm{Enc}(m) = C$,$\mathrm{Dec}_{HE}(\mathrm{Enc}(sk), C) = C' = \mathrm{Enc}(m)$.

CKKS

- CKKS[2] - FHE scheme for approximate arithmetic
- Supports both approximate addition and approximate multiplication on encrypted data
- Security relies on hardness of RLWE problem
- Enable interesting application, e.g. privacy-preserving machine learning

Contents

CKKS



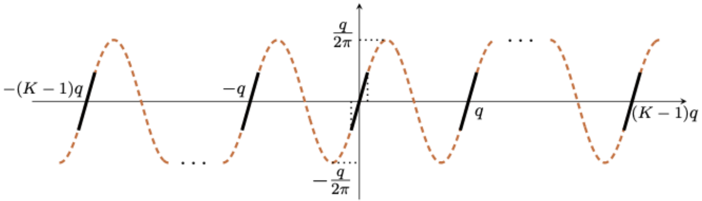
$\mathbb{C}^{N/2}$
Message
m
encode
$\mathbb{Z}[X]/(X^N+1)$
Plaintext
$p(X)$
encrypt
$(\mathbb{Z}_q[X]/(X^N+1))^2$
Ciphertext
$c = (c_0(X), c_1(X))$
compute f
Ciphertext
$c' = f(c)$
$(\mathbb{Z}_q[X]/(X^N+1))^2$
decrypt
Plaintext
$p' = f(p)$
$\mathbb{Z}[X]/(X^N+1)$
decode
Message
$m' = f(m)$
$\mathbb{C}^{N/2}$

CKKS

- Messages are vectors in $\mathbb{C}^{\frac{N}{2}}$
- recall: RLWE problem
- Before encryption, vectors are encoded as plaintexts in integer polynomial ring $\mathbb{Z}[X]/(X^N - 1)$
- Enable ‘batching/vectorisation’
- With this encoding, CKKS use RLWE as building block, instead of LWE, s.t. CKKS benefits from a smaller size of public key and faster multiplication.
- (LWE: public key is of size $\mathcal{O}(n^2)$, multiplication is in $\mathcal{O}(n^2)$)
- (RLWE: public key is of size $\mathcal{O}(n)$, multiplication is in $\mathcal{O}(n \log n)$ (with FFT))

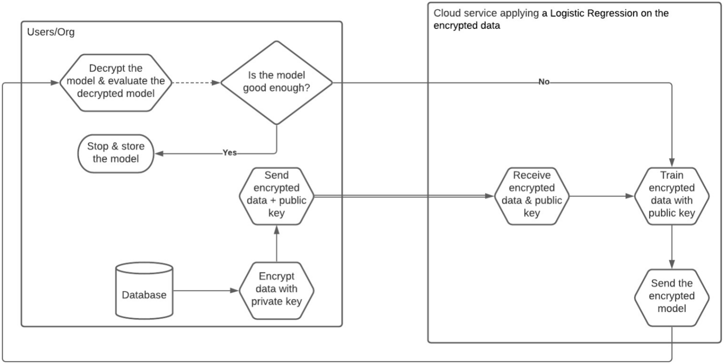
Bootstrapping of CKKS

- Bootstrapping of CKKS scheme includes a homomorphic modular reduction
- Cheon et al. [3] proposed a bootstrapping for CKKS scheme where decryption formula is approximated by an approximate polynomial of a scaled sine function
- A trigonometric function is a good approximation of modular reduction because it is the identity nearby zero and periodic with period q



Use Case: Privacy-Preserving Machine Learning

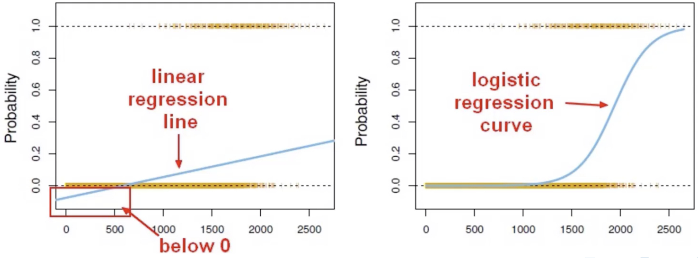
- Users encrypt sensitive data, then send them to server
- Cloud service trains machine learning model with encrypted data
- Cloud service sends the encrypted, tranined model back to users



Contents

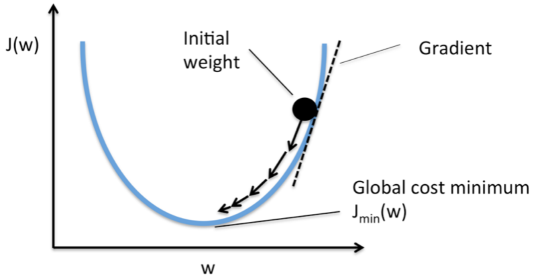
Logistic Regression

- Supervised learning
- 2-class classification
- Model the posterior probability as logistic sigmoid function:
  $Pr(Class = 0|input = x) = \frac{1}{1+e^{-w^T x}} = \sigma(w^T x)$
- Sigmoid function always has value in [0, 1]
- Logistic regression is more suitable for predicting binary variable
- Linear regression is more suitable for predicting continuous variable, may predict value outside of [0, 1]



Gradient Descent

- Go opposite direction of the gradient of the current point of the function/minimum of the function
- In machine learning: minimize the error of the model - minimum of the loss function
- For every training epoch, update the weight as follows:
  $w^{(\text{new})} = w^{(\text{old})} - \alpha(\sigma(Xw) - t)X^T$,
  where α is the learning rate, $\sigma(Xw)$ is the predicted value from the sigmoid function, and t is the target value.



Train a Logistic Regression Model on Data Encrypted by CKKS

- recall: CKKS supports homomorphic addition and multiplication
- recall: sigmoid function: $\sigma(w^T x) = \frac{1}{1+e^{-w^T x}}$
- approximate polynomial for the sigmoid function as follows: $\sigma(x) = 0.5 + 0.197x - 0.004x^3$

References

C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, ser. STOC '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 169–178. [Online]. Available: https://doi.org/10.1145/1536414.1536440

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.

J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song, "Bootstrapping for approximate homomorphic encryption," in *Advances in Cryptology – EUROCRYPT 2018*, J. B. Nielsen and V. Rijmen, Eds. Cham: Springer International Publishing, 2018, pp. 360–384.