Homomorphic Machine Learning with SEAL

Hao Chen[1], Kyoohyung Han[2], Zhicong Huang[3], Amir Jalali[4], Kim Laine[1], Ran Gilad-Bachrach[1], Kristin Lauter[1]

[1]Microsoft Research, [2]Seoul National University, [3]École Polytechnique Fédérale de Lausanne, [4]Florida Atlantic University

Homomorphic Machine Learning

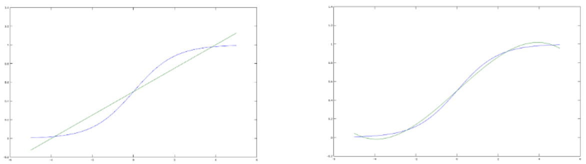
To run learning phase of machine learning in encrypted state, what is the problem? **message size problem**

Data: X, y
Result: W
α : learning rate, N : number of samples, K : number of variables;
initialization W (as origin point);
initialization r (as origin point);
for i *in* $[0, N)$ **do**
  compute inner product $\langle \vec{X}_i, \vec{W} \rangle = V_i$;
  compute approximate sigmoid f to V_i;
end
for i *in range* $[0, K)$ **do**
  compute derivative $\Delta_i = \sum_{j \in B_k} (Y_j - f(V_j)) \cdot X_{j,i}$;
  add residue to derivative $\Delta_i = \Delta_i + r_i$;
  extract sign of derivative $\mathsf{sign} = 1$ if $\Delta_i > 0$ otherwise -1;
  update weight vector element $W_i = W_i + \alpha \cdot \mathsf{sign}$;
  update residue vector element $r_i = \Delta_i + \alpha \cdot \mathsf{sign}$;
end

Instead of using gradient decent, "1-Bit Stochastic Gradient Descent" used only sign information in learning and this solves the message size problem [1].

Polynomial Approximate of Sigmoid Function

- **Mini-max polynomial approximate**
  - Range [-5, 5]
  - Degree 1 and 3 for each

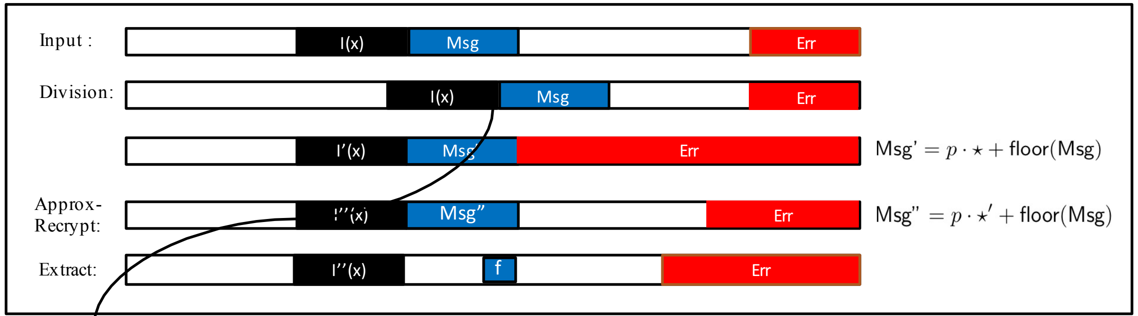


(a) $f(x) = 0.5 + 0.125x$ (b) $f(x) = 0.5 + 0.197x - 0.004x^3$

Figure 1: Mini-max approximate for sigmoid

Homomorphic Flooring using SEAL

FV style encryption of message m: $c_1(x) + c_2(x) \cdot sk(x) = q \cdot I(x) + \Delta \cdot \mathsf{msg} + \mathsf{err}$



$$(c_1(x), c_2(x)) \Rightarrow \frac{q}{p^r} \cdot m(x) + e(x) + q \cdot I(x)$$

$$\left(\frac{c_1(x)}{p^{r-1}}, \frac{c_2(x)}{p^{r-1}}\right) \Rightarrow \frac{q}{p^r} \cdot \frac{m(x)}{p^{r-1}} + \frac{e(x)}{p^{r-1}} + \frac{q}{p^{r-1}} \cdot I(x)$$

$$= \frac{q}{p^r} \cdot \left(p \cdot I(x) + \left\lfloor \frac{m(x)}{p^{r-1}} \right\rfloor\right) + [m(x)]_{p^{r-1}} + \frac{e(x)}{p^{r-1}}$$

- Only need to keep lowest digit part, this makes our recrypt faster with approx-recrypt.
- Full process is little-bit expensive than bootstrapping, but we can perfume bootstrapping and flooring at once.
- Sign Extraction can be expressed by floor function.
- Furthermore, we can adapt SIMD technique.

Encryption Parameter

- We used SEAL with RNS-FV version and bootstrapping implementation.
- Each coefficient modulus chain is 60-bit prime integer

$\texttt{coeff\_mod\_count} = 16$
$\texttt{poly\_modulus} = X^{32768} + 1$
$\texttt{plain\_modulus} = 2,048,383$

Setting and Result

- **Number of samples: 1000 ~ 2000**
- **Number of features: 10 ~ 64**
- **Number of iteration: 10 ~ 20 (depend on learning rate)**

[1] Seide, Frank, et al. "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech dnns." *Fifteenth Annual Conference of the International Speech Communication Association*. 2014.