Differentially Private Machine Learning under FHE

Revisiting ML Training under FHE: Convergence Guarantees, Differential Privacy, and Efficient Algorithms

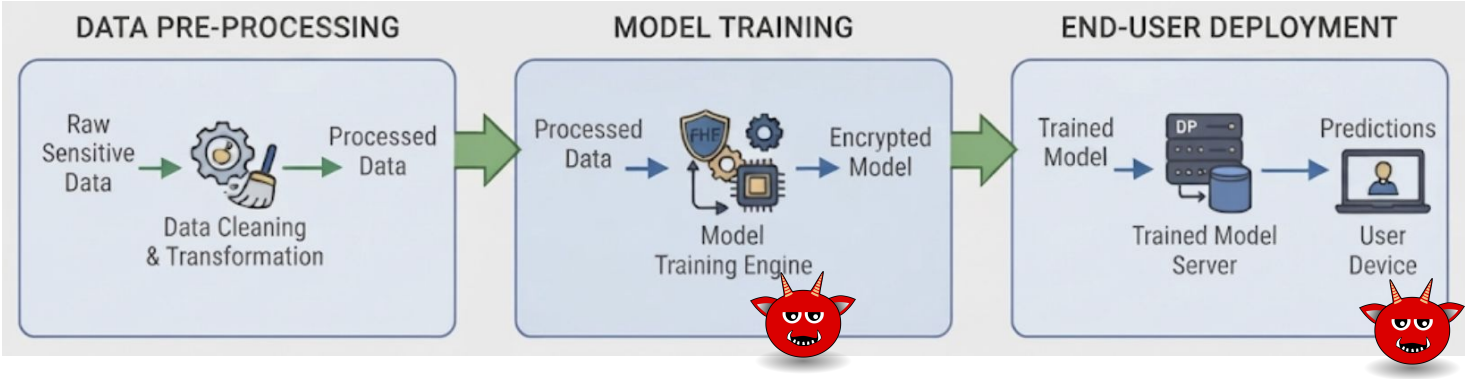
Yvonne Zhou[1](Presenter), Mingyu Liang[1], Ivan Brugere[2], Danial Dervovic[2], Antigoni Polychroniadou[2, 3], Guo Yue[2, 3], Min Wu[1], Dana Dachman-Soled[1]

[1]University of Maryland, [2]JP Morgan AI Research, [3]AlgoCRYPT CoE

October 26, 2026 @ NIST WPEC 2026

Problem Statement

- **Privacy concern in ML** sensitive domains (e.g., finance, healthcare)
- Many clients **lack the resources** (compute, memory, time) to train models locally → must **outsource training** to powerful external servers
- Privacy risks in ML pipeline:

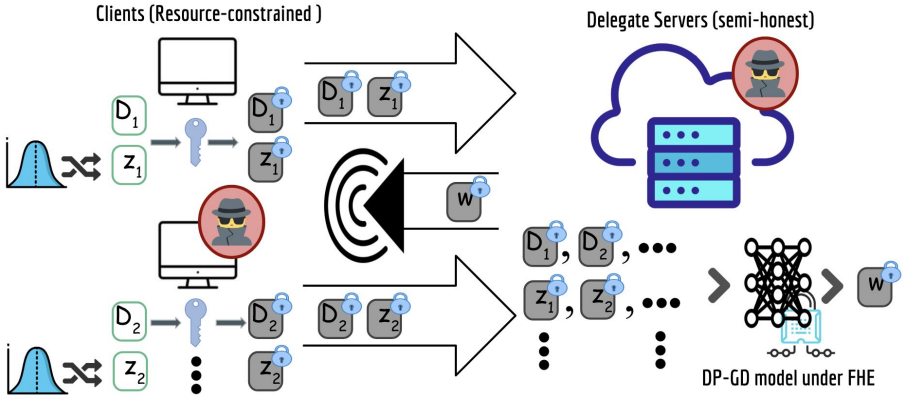


Goal: Enable clients to benefit from outsourced training while maintaining **end-to-end privacy guarantees**

Our Approach:

Combine:

- Fully Homomorphic Encryption (**FHE**)
  - ➢ Clients encrypt training data.
  - ➢ Ensures that neither the server nor other clients ever see any raw inputs
  - ➢All computation on encrypted data.
- Differential Privacy (**DP**)
  - ➢ DP is applied to the training process
  - ➢ limit what can be inferred from the output model.



*Extends to a **multi-clients** setting using cryptographic tools such as multi-key FHE*

Incorporating Differential Privacy (DP)

- Even with encrypted training, the final model (once decrypted) may still leak information
  - Ex: Model Inversion — Reconstructing a Face from facial recognition model [Fredrikson et al., 2015]



model inversion attack



training set image

Comparison to Prior Work

- Existing FHE-based training, e.g., [Han et al., 2018]:
  - Only provide experimental evaluation of their FHE approach
  - No principled method to select polynomials or define hyperparameters
    ⇛ **Our work**: Provide convergence theory,
    Guideline for selecting hyperparameters

- FHE + DP work. e.g.,[Sébert et al., 2023]:
  - Encrypts model updates (e.g. gradients)
  - Requires interactive rounds between between client and server
    ⇛ **Our work**: Encrypts data, and lets the server run the full training non-interactively

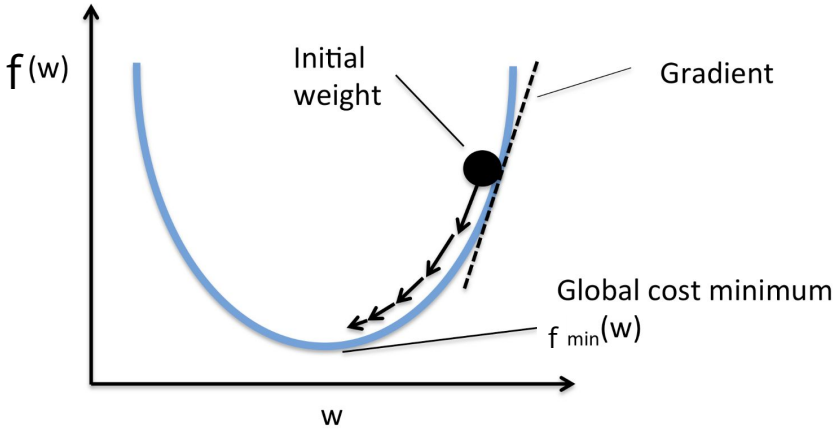
- DP outsourcing, e.g., [Dagher et al., 2020]
  - Focuses on query release, not full ML-training
    ⇒**Our work**: full pipeline for DP-ML training under FHE

Background:

- **Gradient descent (GD)** is the standard training algorithm for loss function f(w):
- Pseudocode of GD:
  - **Objective:** minimize a function $f(w)$
  - **Initialization:** choose an initial weight w_0
  - **Iteration:** for $t = 0, 1, \ldots, T-1$,
    - compute the gradient $\nabla f(w_t)$
    - update the weight

$$w_{t+1} = w_t - \text{learning_rate} \cdot \nabla f(w_t)$$

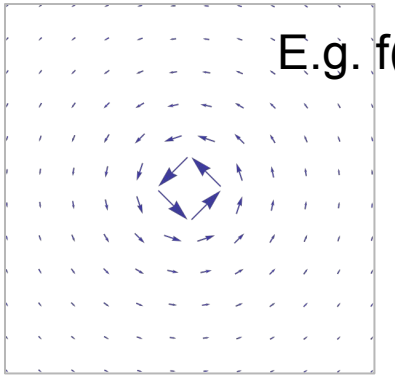
  - **Output:** final weight w_T



Background (cont'd):

- GD computes gradients through **non-linear** activations (e.g., Sigmoid $\sigma(x) = \frac{1}{1+e^{-x}}$)
- FHE supports only **+ and ×**
- Replace nonlinear activation with polynomials P.
- During training, compute an approximate gradient: ∇ g(w) ≅ ∇ f(w)
- Q1: Is ∇g(w) conservative?
  → Does there exist an objective g(w)?

  Easy Ans: For the case of neural nets, ∇ g(w) is gradient of some function g(w)



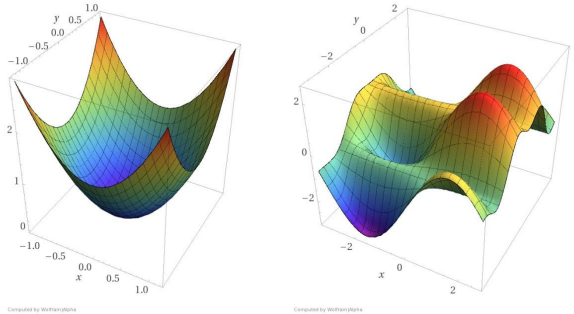
E.g. f(x,y)=(−y,x)

- Result: training is done on **approximate objective** g(w), where,

$$\forall \mathbf{w} \in \mathcal{D}, ||\nabla f(\mathbf{w}) - \nabla g(\mathbf{w})|| \leq \zeta$$

Theoretical Challenge: Convergence of Approximate GD

- Even if $\nabla f \cong \nabla g$, g may not preserve convexity
  - g is polynomial, may have lot of local minima
- Standard convergence guarantees (e.g., GD or DP-GD) do not apply



- Question?
  - Can gradient descent on g still converge to (or near) the optimum of f?

Our Contribution:

- First convergence analysis of GD on polynomial approximations under FHE
- Select better polynomial approximation (e.g minimax vs least-square error)
- Enables data-independent selection of learning rate and training steps

Convergence of Approximate gradient descent (informal)

- Assume: gradient approximation error bounded by ζ
- g is β-smooth, and the true objective f is μ-strongly convex.
- Use a learning rate: η= 1/β, after sufficient steps, $T = \frac{2\beta_g(g(\mathbf{w}_{g,0}) - g(\mathbf{w}_g^*))}{\rho^2}$,
- The output of g remains within a bounded distance of the true optimum:
  - Increasing in gradient approximation error ζ
  - Decreasing in strong convexity μ.

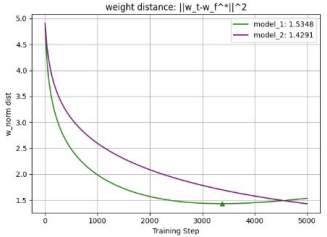
$$||\mathbf{w}_{g,T} - \mathbf{w}_f^*||^2 \leq \frac{4\alpha}{\mu} + \frac{\rho^2 + 2\zeta\rho + \zeta^2}{\mu^2}.$$

- **Observations** from theorem:
  - T, required for convergence, grows with the smoothness parameter β_g .
  - Approximation error ζ, controls final optimization error.
  - Empirically compare two approximation methods: ***minimax***[Remez, 1934] vs ***least-square*** [Björck, 1996]

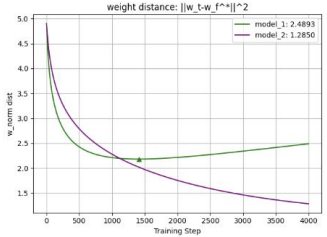
Empirical Validation of the Convergence Guarantee

- Smaller β → fewer iterations to get ρ, where $\|\nabla g(\mathbf{w}_{g,\tilde{t}})\| \leq \rho$
- Smaller approximation error ζ → closer to the true optimum

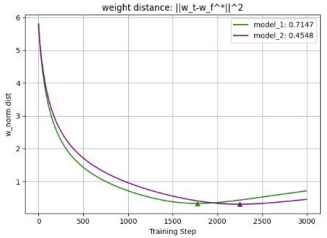
Dataset	Method	Smoothness	Steps@ρ	Error
MNIST	MM@[-20, 20]	19.1185	2650	0.1920
	LS@[-20, 20]	35.1523	4873	0.1439
	MM@[-25, 25]	15.3325	2125	0.2335
	LS@[-25, 25]	27.4613	3806	0.1834
	MM@[-10, 10]	2.3223	1287	0.0632
	LS@[-10, 10]	2.5594	1419	0.0498
Adult	MM@[-15, 15]	1.6680	924	0.1356
	LS@[-15, 15]	2.6682	1479	0.0955



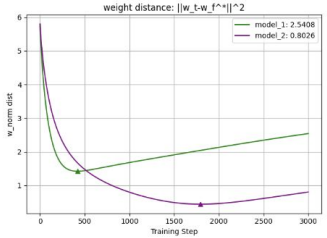
(a) Fig a-1: Mnist Pair-1



(b) Fig a-2: Mnist Pair-2



(d) Fig b-1: Adult Pair-1

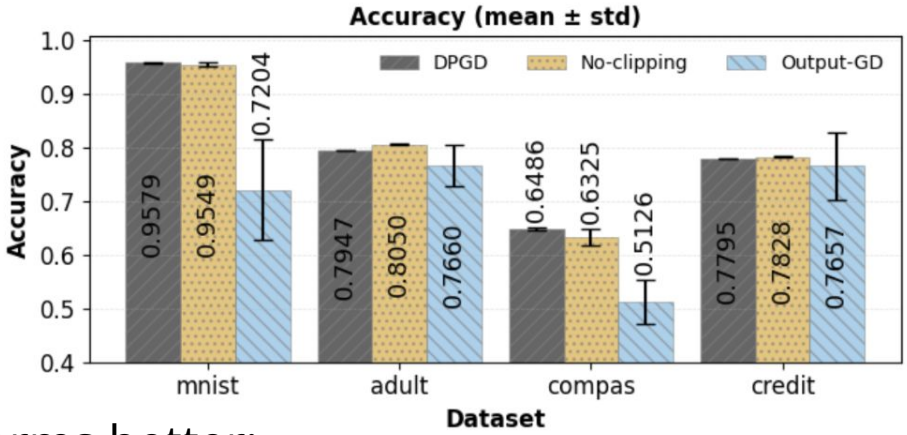


(e) Fig b-2: Adult Pair-2

Compare two models' converge rate using 7th degree polynomials derived from minimax vs least-square, converge to ρ<0.1 for mnist, and ρ<0.05 for adult, where

Achieving DP in Gradient Descent

- Two main methods in the literature:
  - **Gradient perturbation**: DP-noise (usually Gaussian) is added to the gradients at each iteration
  - **Output perturbation**: Train a no-DP model first, then add noise to the final model parameters.



- Gradient perturbation typically performs better:
  - A **single large** noise vs **distributing noise** across iterations
  - Empirically shows: Output-perturbation models exhibit **low accuracy**, with **high variance** across runs

Therefore, in our work we focus on how to achieve DP via **gradient perturbation** under FHE.

Challenges to Guarantee Differential Privacy (DP)

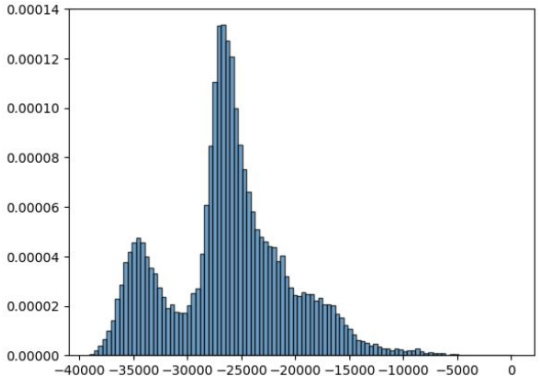
- Mechanism: Add **gaussian noise** during training (DPGD)
- Standard DP requires **per-sample gradient clipping** to bound sensitivity

$$\text{Clip}(\nabla f_{(\mathbf{x}_j||y_j)}(\mathbf{w}), C) = \min\{1, \frac{C}{||\nabla f_{(\mathbf{x}_j||y_j)}(\mathbf{w})||}\}\nabla f_{(\mathbf{x}_j||y_j)}(\mathbf{w})$$

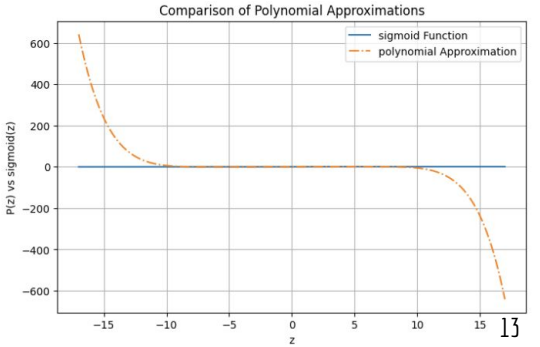
- Problems 1: Clipping is **expensive** under FHE due to complex operations
  - **L2 norm** of the gradient
  - **Compare (or max)** the norm to the clipping threshold
  - Scale the gradient with **inverse multiplication**

Challenges to Guarantee Differential Privacy (DP)

- Problems 2: Breaks DP guarantees, because the gradient sensitivity cannot be rigorously bounded.
  - Polynomial approximations can only achieve bounded error on a **closed interval**.
  - Outside the interval error can be **unbounded**.
    - e.g: Approximation of $\phi(< w, x >)$ only accurate if we have a bound on $|<w, x>|$. (*bottom figure*)
  - In all prior works using polynomial approximations these bounds are chosen **heuristically**
  - This problem is inherent to any method utilizing polynomial approximations while trying to achieve DP, not just gradient clipping



(c) z values at t=1086



Efficient DP under FHE

Our contribution:

- We design an FHE-compatible DP training algorithm:
  - Modify both the objective function and the training procedure to remain within FHE-friendly operations
  - Avoids per-sample gradient clipping
  - Achieves formal DP guarantees with significantly **lower computational overhead**
- We offer the first comprehensive **convergence analysis** of differentially private training under approximate gradient descent.

Non-clipping DP-GD approach

1. Modified objective function with **barrier function** (more on next slide):

$$f_{\kappa}^{+B}(\mathbf{w}) := f(\mathbf{w}) - \lambda B_{\kappa}(\Theta - \|\mathbf{w}\|^2),$$

2. Bound the **gradient sensitivity** (more on next slide)
3. Compute the noise scale, using the **zCDP**-based composition (for full dataset training), to preserve (ε, δ)-DP as:

$$\text{set } \sigma = \frac{2\Delta_2\sqrt{T\log(3/\delta)}}{\epsilon N}, \Delta_2 = 2(\phi'_{max} + e_f)\sqrt{m}$$

Barrier Function

1. Modified objective function with **barrier function**:

$$f_{\kappa}^{+B}(\mathbf{w}) := f(\mathbf{w}) - \lambda B_{\kappa}(\Theta - \|\mathbf{w}\|^2),$$

- Barrier function $-\lambda B_{\kappa}(\Theta - \|w\|^2) = -\lambda \ln(\Theta - \|w\|^2)$ approaches ∞ as $\|w\|^2$ approaches Θ
- Typically used as a way to control magnitude of the optimum w
- We replace $\nabla B_{\kappa} = 2\lambda \frac{1}{\Theta - \|w\|^2} w$ with polynomial approximation of 1/x
- Parameterized by κ: We will require our polynomial approximation of 1/x to achieve bounded error only on the interval [$\kappa\Theta$, Θ].
    - As $\kappa \to 0$, becomes harder and harder to approximate 1/x with small error

Gradient Sensitivity

2. Bound the **gradient sensitivity**

$$f_\kappa^{+B}(\mathbf{w}) := f(\mathbf{w}) - \lambda B_\kappa(\Theta - \|\mathbf{w}\|^2),$$

- Derive an explicit upper bound on the iterates // w_i // when running GD with the poly-approximated gradient, assuming only that poly approximations satisfy some error bound.
- Given this, can bound |< w, x >| ≦ ||w||*||x||
- This allows us to appropriately choose the approximation interval for ∇ f and instantiate with a specific polynomial approximation achieving the error bound
- Using the bound on the gradient sensitivity of the original objective plus the bound on the approximation error, we obtain a bound on the sensitivity of the poly-approximated gradient

Multiplicative Depth

- Removes clipping with a lightweight "Barrier Calculation"
- Running **once** per iteration
- Can be run in **parallel** with gradient calculation
- Depth per iteration drops from **24 to 9**.
- Same as no-DP training

Circuit for Logistic Regression Model

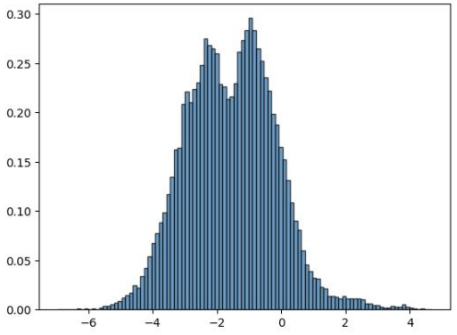
Computational Step	DP-GD (w/ Clip)	DP-GD (No Clip)	Output-GD
Path A: Gradient Calculation (Per-sample)			
$\mathbf{z} = \langle \mathbf{w}, \mathbf{x} \rangle$ (Inner Product)	+1	+1	+1
Sigmoid Poly	+4	+4	+4
Gradient $((y_{pred} - y) * \mathbf{x})$	+2	+2	+2
— *Per-Sample Clipping* —			
$\|\|\text{grad}_i\|\|^2$ (Norm)	+1	—	—
$\sqrt{\text{norm}}$ (Sqrt Poly)	+4	—	—
$\max\{C, \|\|\text{grad}_i\|\|\}$	+6	—	—
Inverse $\sqrt{\text{norm}}$ (using TS)	+3	—	—
Scaling Logic (grad_i*scal)	+1	—	—
Average Gradient	+1	+1	+1
Path B: Barrier Calculation (Parallel to A)			
$\|\|\mathbf{w}\|\|^2$ (Squared Norm)	—	+1	—
Inverse Term (using LS)	—	+3	—
Barrier Term	—	+2	—
Critical Path Depth	23	$\max(8, 6) = 8$	8
Weight Update (grad * lr)	+1	+1	+1
Total Cost per Iteration	**24**	**9**	**9**

Experiments Results

DP-(S)GD, vs Output-perturbation, vs Our method in 128-bits Security

- **Accuracy**: accuracy outperformance, and AUC drop max a 0.33% over DP-GD
- **Speed**: At least **5× speedup** over DP-GD, match to no-DP training.
- **DP**: Explicitly accounts for approximation errors, so yields provable DP guarantees

linear hypothesis value remained **bounded** throughout the whole training process



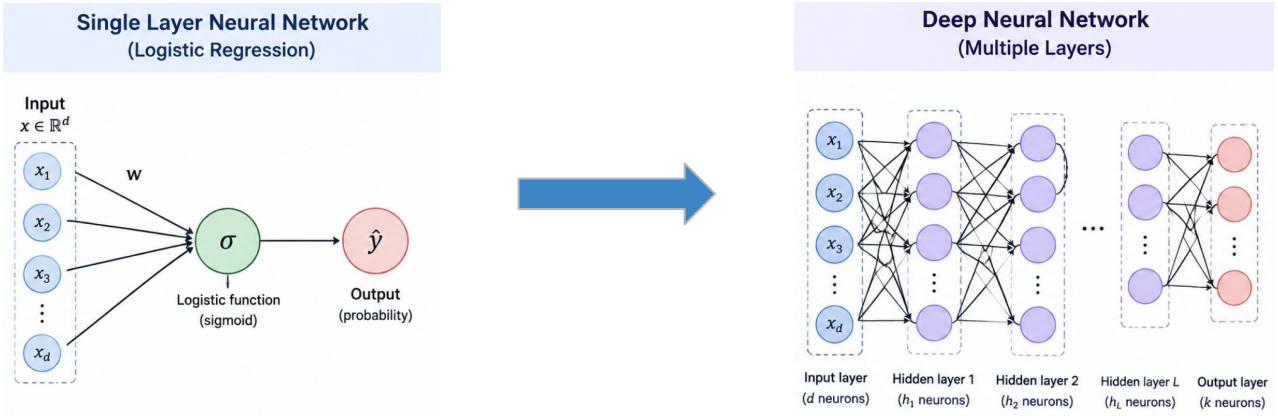
(b) z values at t=1086

Data	Training Model	ACC	AUC	10-threads	20-threads	30-threads	50-threads
mnist	Algo 4 (DP-SGD)	93.77%	98.20%	7996.8 sec/iter	4240.0 sec/iter	3504.4 sec/iter	1813.5 sec/iter
	Algo 5 (No Clip)	93.97%	98.20%	566.7 sec/iter	442.2 sec/iter	409.8 sec/iter	350.5 sec/iter
	Output-SGD	73.87%	86.68%	524.5 sec/iter	434.9 sec/iter	397.4 sec/iter	351.1 sec/iter
credit	Algo 4 (DP-SGD)	77.96%	69.18%	8028.1 sec/iter	3910.5 sec/iter	3236.3 sec/iter	1843.3 sec/iter
	Algo 5 (No Clip)	77.99%	68.85%	521.9 sec/iter	449.2 sec/iter	394.1 sec/iter	343.1 sec/iter
	Output-SGD	76.76%	55.61%	551.4 sec/iter	415.8 sec/iter	379.9 sec/iter	342.0 sec/iter

Summary of contributions

- First **theoretical** convergence guarantees for training with approximate gradient descent under FHE
- First scalable FHE-compatible DP-ML training **algorithm,** with rigorous DP guarantee**,** and significantly computational cost reduction.
- Additionally, we offer **convergence analysis** of the DP-ML training under FHE.
- Building on our convergence analysis, guidelines for optimal **polynomial approximation**
- And data-independent **hyperparameter** tuning

In Progress: Extending to DNNs



Key challenges:

- Bound gradient sensitivity for each layer to calibrate DP noise.
- Bound each activation's input range to control polynomial approximation error under FHE.
- Loose bounds in earlier layers can lead to rapidly expanding input ranges in later layers
- Analyzing optimization behavior is challenging for non-convex DNN objectives. (e.g. local minima, saddle points)

Thank You!

Reference

- Fredrikson et al., 2015] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In Proceedings of the 22nd ACM SIGSAC conference on computer and communications security, pages 1322–1333, 2015
- [Han et al., 2018]:Kyoohyung Han, Seungwan Hong, Jung Hee Cheon, and Daejun Park. Efficient logistic regression on large encrypted data. Cryptology ePrint Archive, 2018
- [Sébert et al., 2023]:Arnaud Grivet Sébert, Marina Checri, Oana Stan, Renaud Sirdey, and Cédric Gouy-Pailler. Combining homomorphic encryption and differential privacy in federated learning. In 2023 20th Annual International Conference on Privacy, Security and Trust (PST), pages 1–7, 2023. doi:10.1109/PST58708.2023.10320195.
- [Dagher et al., 2020]: G.G. Dagher, B.C.M. Fung, N. Mohammed, and et al. Privacy-preserving data outsourcing framework with differential privacy. Knowledge and Information Systems, 62:1923–1960, 2020. doi: 10.1007/s10115-019-01405-7
- [Remez, 1934]E. Ya. Remez. Sur la détermination des polynômes d’approximation de degré donné. Comm. Soc. Math. Kharkov, 10:41, 1934
- [Björck, 1996]Åke Björck. Numerical Methods for Least Squares Problems. SIAM (Society for Industrial and Applied Mathematics), 1996. doi: 10.1137/1.9781611971484.
- Yu-Xiang Wang, Borja Balle, and Shiva Prasad Kasiviswanathan. Subsampled rényi differential privacy and analytical moments accountant. In The 22nd international conference on artificial intelligence and statistics, pages 1226–1235. PMLR, 2019