Federated Learning with Secure Multiparty Computation

Student - Palash Jain
Faculty Advisor - Prof. Minseok Kwon
Summary

- Protect sensitive information of individual/organization while building models using federated learning.
- Make federated learning more scalable using Multiparty computation.
- Use differential differential privacy to secure models from revealing sensitive information.
Summary Milestone 1

- **Reading papers / Books**
  - How to Share a Secret - Adi Shamir
  - Our Data, Ourselves: Privacy via Distributed Noise Generation
  - A Pragmatic Introduction to Secure Multi-Party Computation
  - Programming Differential Privacy

- **Topics**
  - Shamir Secret Sharing
  - Additive Secret Sharing
  - Differential Privacy
  - Federated Learning
  - Multiparty Secure Computation

- **Tools Setup**
  - MPyC
  - Lightweight MPC Framework for Python
Milestone 2 Goals

- Task 1 - Understand and dive deep into the code for Lightweight MPC Framework for Python.
- Task 2 - Implement the distributed random noise function.
Algorithm

1) Each player shares a random bit, 0 or 1.

2) Assume we have a public source of unbiased bits. XOR c bits with b to convert low quality b bits into high quality bits.

3) Replace the shares s by 2s - 1.

4) Sum each participants sum to get share of binomial noise.
Challenges faced

- Minimal documentation present
- Understanding the implementation in the paper
- Generating public coins
- Making the distribution close to gaussian distribution
Milestone 3 Tasks

- Migrate the noise implementation to MPC infrastructure.
- Test results of federated learning.
Thanks