Secure Multi-party Computation in the Context of Deep Learning

CS6160 : Cryptology

Raj Patil : CS18BTECH11039

2021-11-22

Outline

Context

- DL is useful
- inference needs to be fast -> quantization
- privacy is important

Secure Multi-Party Computation

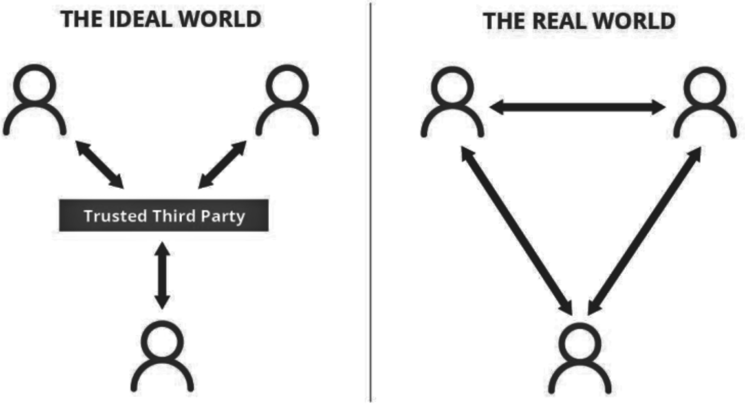

Figure: Incentive for MPC protocols

Requirements of MPC

1. Input Privacy
2. Correctness

Deconstructing the Problem

Really 2 orthogonal sub-problems

1. quantizing neural networks
2. facilitating secure inference

Canonical Perspective of Neural Networks

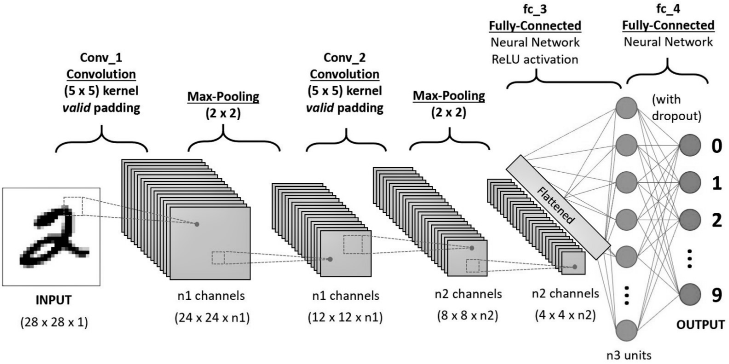

Figure: a typical CNN

A Basic 2PC Framework

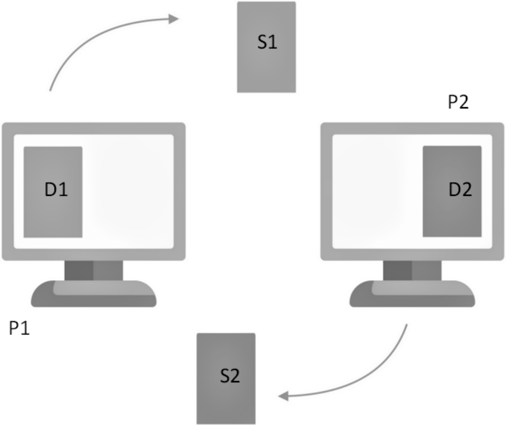

Figure: basic 2PC framework

Secure 2-Party Addition

The two parties (P_1 and P_2) want to securely compute $DS_1 + DS_2$.
Now:

$$\begin{aligned} DS_1 + DS_2 &= (D_1 + S_1) + (D_2 + S_2) \\ &= (D_1 + S_2) + (D_2 + S_1) \end{aligned}$$

```
func splitter(DS):
    S <- get_rand() \\ randomly sourcing the share
    D <- DS - S     \\ the share doesn't leak any
    return D,S      \\ information about DS
```

Approximating Complex activation functions

Two kinds of irregularities:

1. A piece-wise differentiable activation function (e.g. ReLU)
2. A inherently transcendental function (e.g. Sigmoid)

Quantization

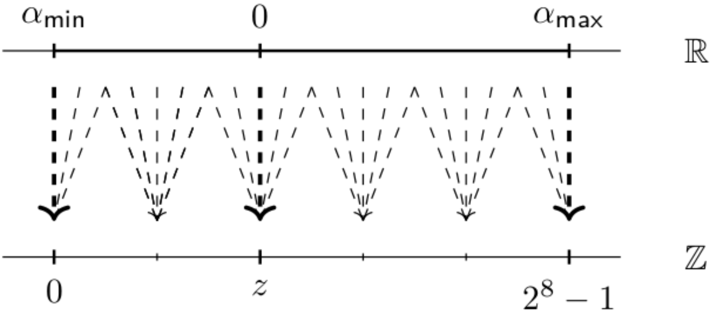

Figure: Quantization

Current Solutions and Challenges

Challenges

- Lack of availability in common frameworks
- Trade-off between efficiency and accuracy during quantization
- Specific Constraints
- Gnosticism regarding efficient solutions -> lack of scrutiny

Contribution

- Probabilistic Truncation (during quantization) :- efficient and accurate
- Model Agnosticism

Testing: Client Server Protocol

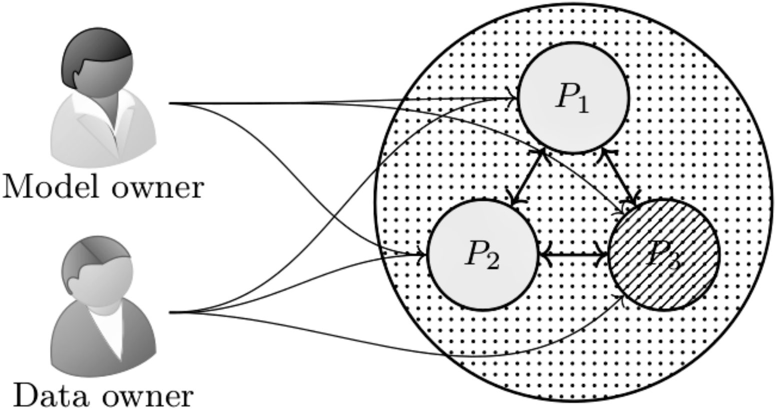

Figure: client server model for inference by a neural network: P_3 is dishonest

Testing: Factors Involved

Table: Possibilities of Adversarial Conditions in MPC protocols

Breakdown of Adversarial Conditions		**Adversarial Nature**	
		Active	*Passive*
Majority	*Honest*	<50% malicious deviants	<50% malicious observers
	Dishonest	>50% malicious deviants	>50% malicious observers

Possibilities

- MPC protocols -> a little too static -> lack of online implementations
- FHE (Fully Homomorphic Encryption) -> inference on encrypted data -> more dynamic opportunities