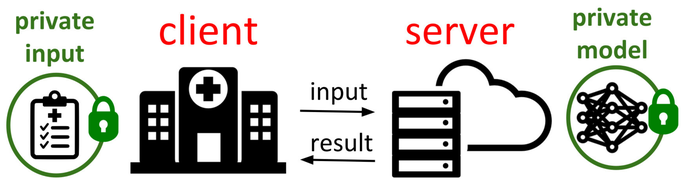
Circa: Stochastic ReLUs for Private Deep Learning

Zahra Ghodsi, Nandan Kumar Jha, Brandon Reagen, Siddharth Garg

Introduction

- Machine learning as a service (MLaaS) gives raise to privacy concerns:
  - Client's input is private and server's model is proprietary

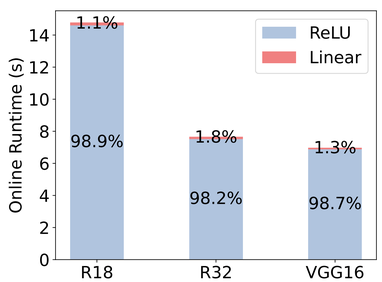


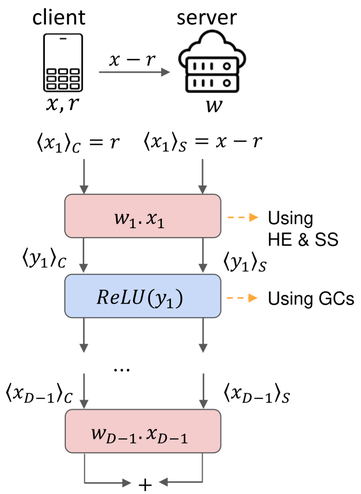
- Private inference (PI) allows the inference computation while protecting the input and model privacy
- PI is based on cryptographic techniques and incurs substantial slowdown

Private Inference

- PI frameworks (e.g., Gazelle [1], Delphi [2]) use different protocols for linear and non-linear layers
- Each layer's values are secret shared between the client and the server
- ReLU operations dominate latency in private domain

PI online runtime for non-linear (ReLU) and linear layers on C10 dataset based on Delphi protocol



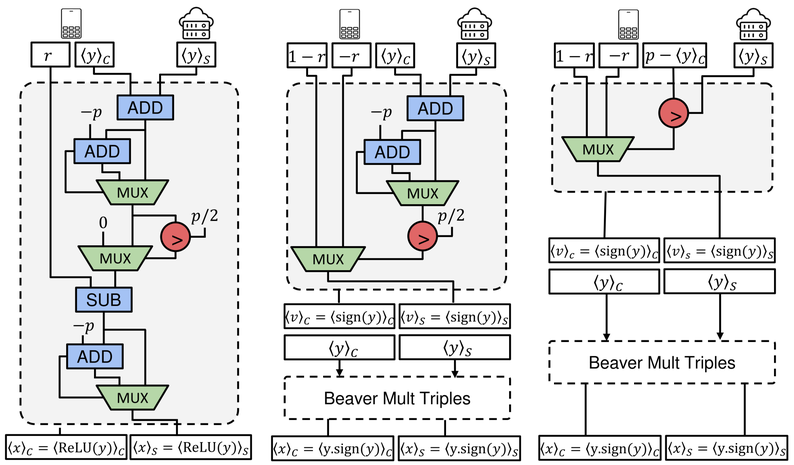


Circa

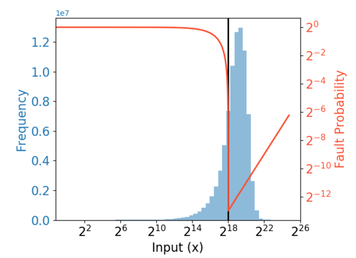
- ReLUs are implemented using Garbled Circuits (GC) with a size of 17.2KB per ReLU (5GB per ResNet32 inference)

▶ Stochastic ReLUs

- Refactoring ReLUs
  - Refactor ReLU(x) as x.sign(x) and implement multiplication with Beaver multiplication triples and sign with GCs
- Stochastic Sign
  - Reduce GC cost by omitting expensive modulo operation and using only a comparator and a MUX
  - Stochastic Sign incurs a fault rate of $P = \frac{|x|}{p}$ (proof in paper)
- Truncated Stochastic Sign
  - The comparison inside GC can be performed over truncated by k-bit inputs
  - Additional fault rate over small values $P = \frac{2^k - |x|}{2^k} \quad \forall x \in [0, 2^k)$ (proof in paper)



Histogram of ResNet18 activations and fault probability at first layer

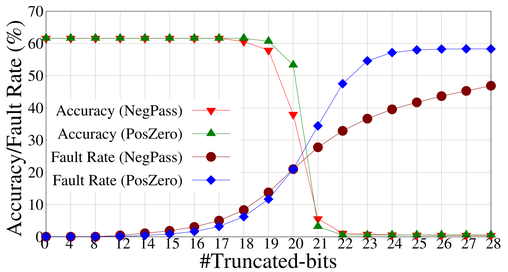


- Small positive activations incur a high fault probability
- Outside this range the prob is small and grows linearly with abs value

Circa Evaluation

- Circa truncates 17-19 bits with $< 1\%$ accuracy loss at fault rate $< 10\%$

ResNet18 on TinyImageNet

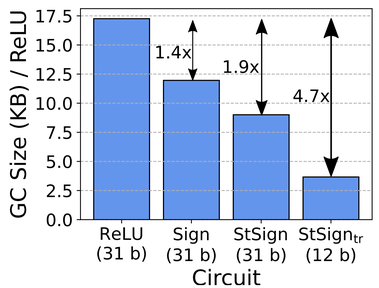


Circa works in two modes of operation:

- Zero out small positive vals (PosZero)
- Pass through small negative vals (NegPass)

Network-Dataset	Baseline Acc (%)	Circa Acc (%)	Baseline Runtime	Circa Runtime	Runtime Speedup
ResNet18-C100	74.24	73.76	11.05	4.15	2.7×
VGG16-C100	73.94	73.25	5.89	2.25	2.6×
ResNet18-Tiny	61.60	60.65	44.55	14.28	3.1×
VGG16-Tiny	50.85	50.73	21.41	6.96	3.1×

- Circa can be applied over *any* pre-trained network without the need to retrain
- Three optimizations from Circa build on top of each other to reduce GC size and PI runtime



References

[1] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "Gazelle: A low latency framework for secure neural network inference," in *USENIX Security*, 2018.

[2] P. Mishra, R. Lehmkuhl, A. Srinivasan, W. Zheng, and R. A. Popa, "Delphi: A cryptographic inference service for neural networks," in *USENIX Security*, 2020.

zghodsi@ucsd.edu https://ghodsi.me