REDsec: Running Encrypted Discretized Neural Networks in Seconds

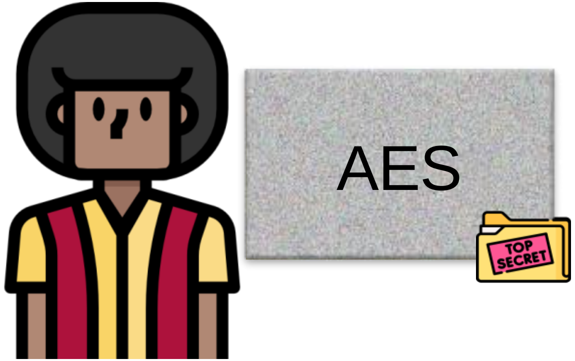
A Fully Homomorphic Approach

Lars Wolfgang Folkerts, Charles Gouert,
Nektarios Georgios Tsoutsos

University of Delaware

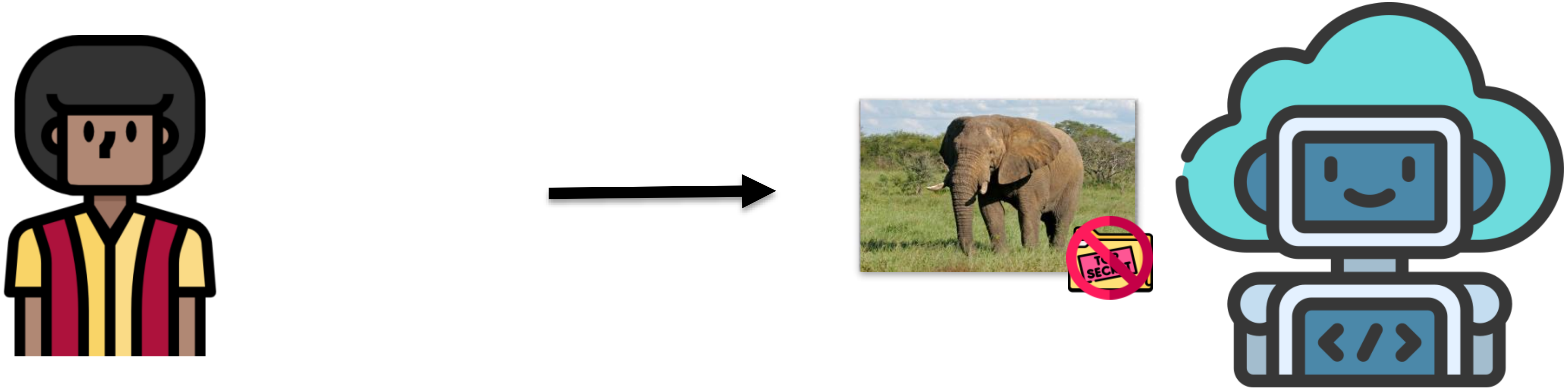
Problem Statement

Machine Learning as a Service



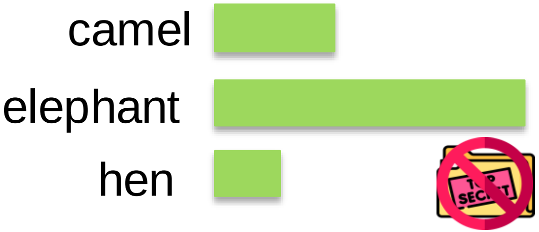
Problem Statement

Machine Learning as a Service



Problem Statement

Machine Learning as a Service

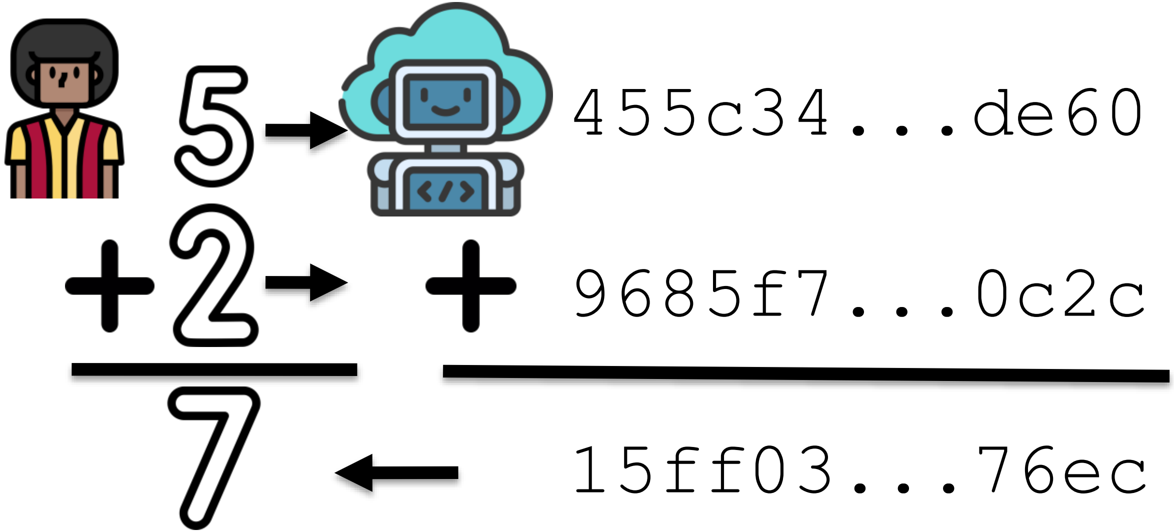


Motivation

Traditional
(e.g. AES)

Privacy

Homomorphic
Cryptography



Motivation

Traditional
(e.g. AES)

Privacy

Homomorphic
Cryptography

5
×2

10

455c34...de60
× 9685f7...0c2c

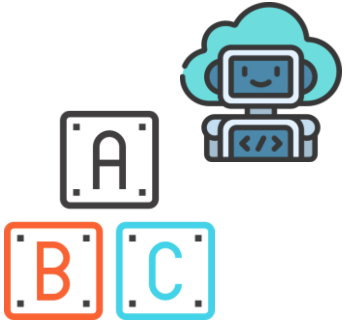
2ad4d6...f401

Motivation

Traditional
(e.g. AES)

Privacy

Homomorphic
Cryptography

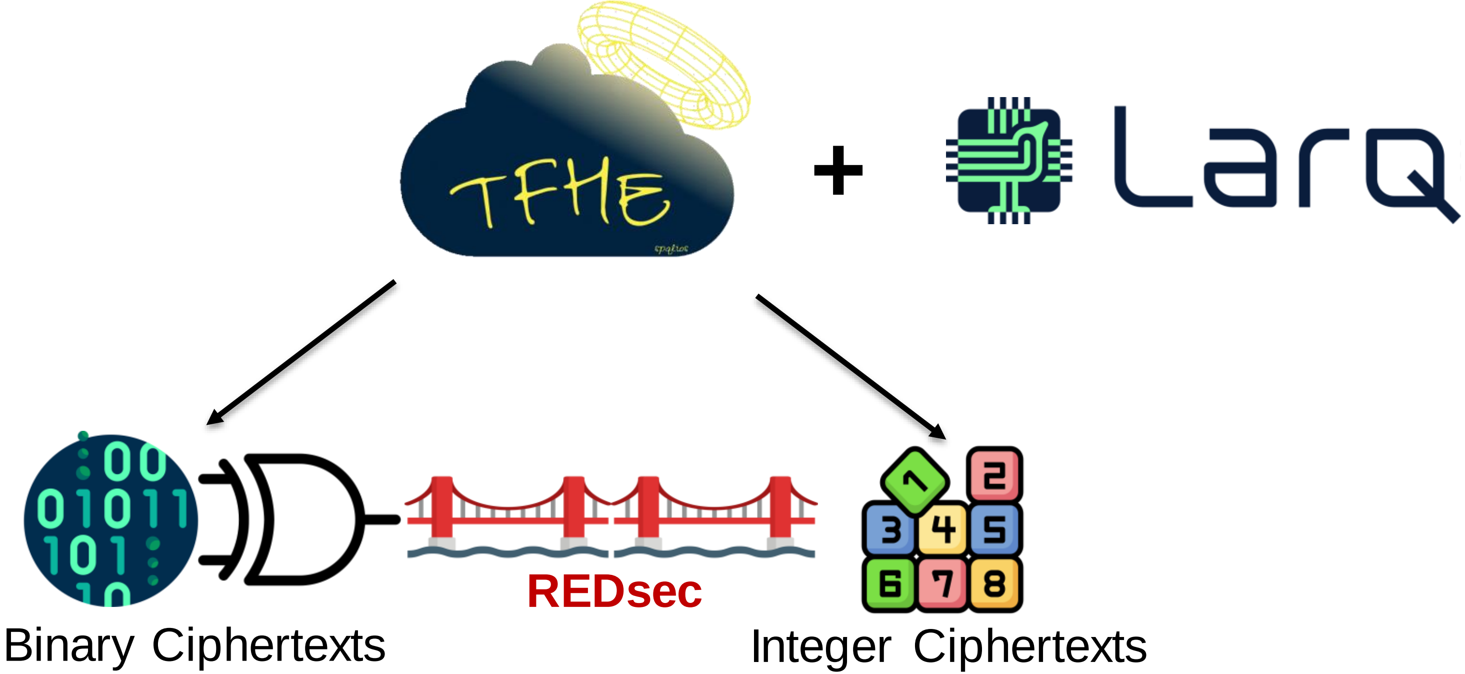


Shallow Problems

Slow

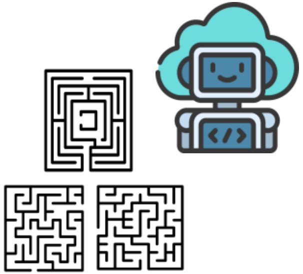
Limited Usability

Our Approach: REDsec



Highlights

Private



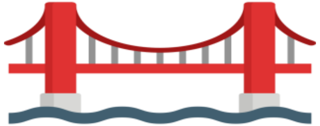
Deep Problems

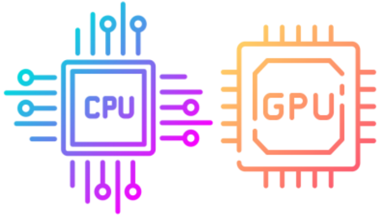
Fast

BYON

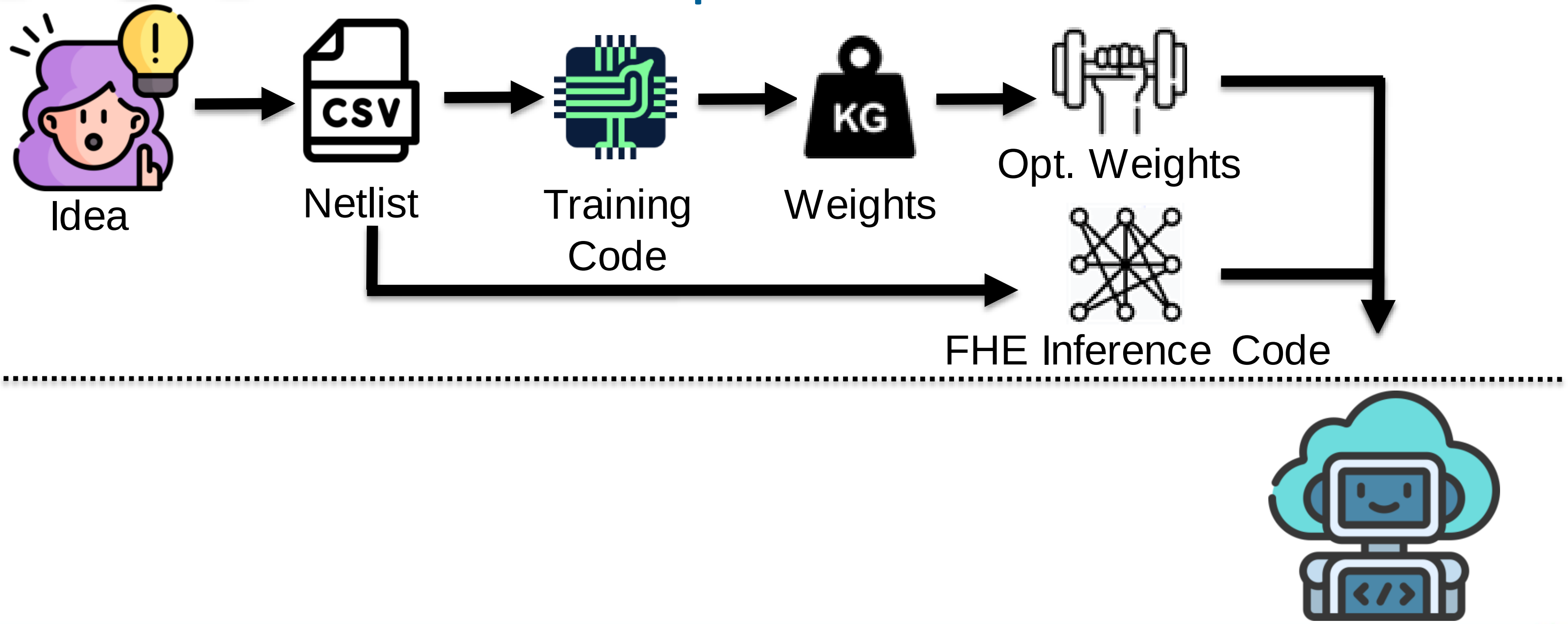
Accessible Framework

IMAGENET

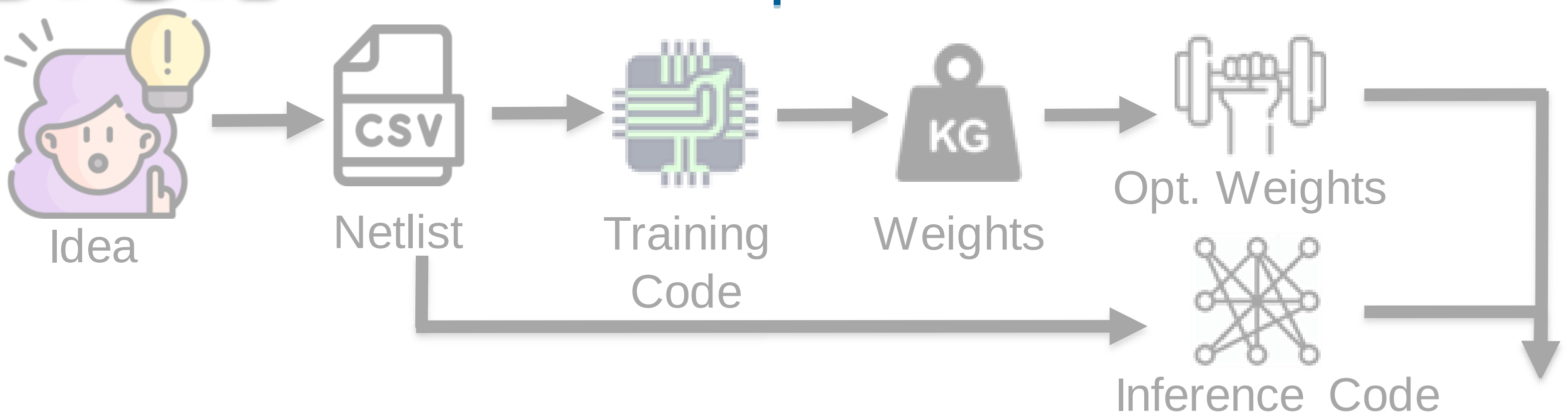


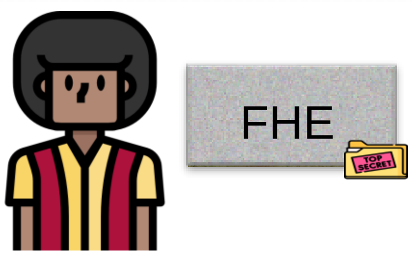


Example scenario



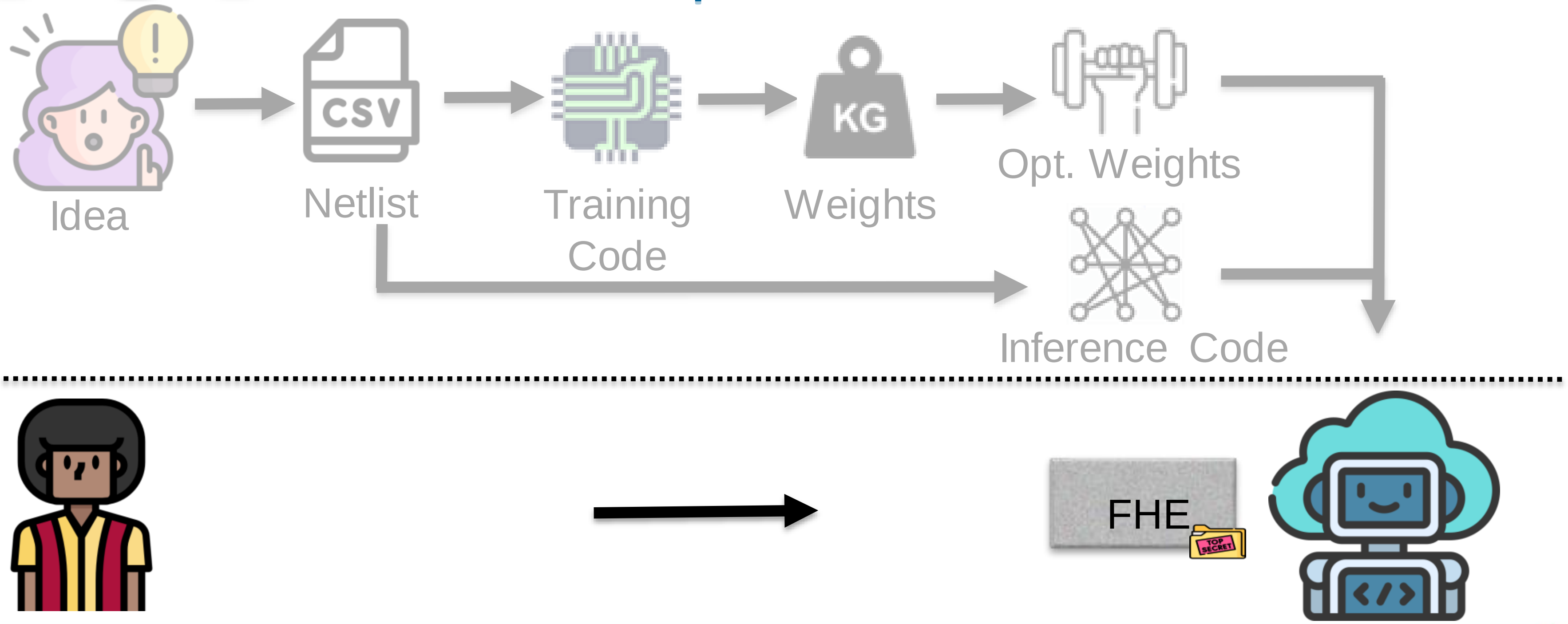
Example scenario





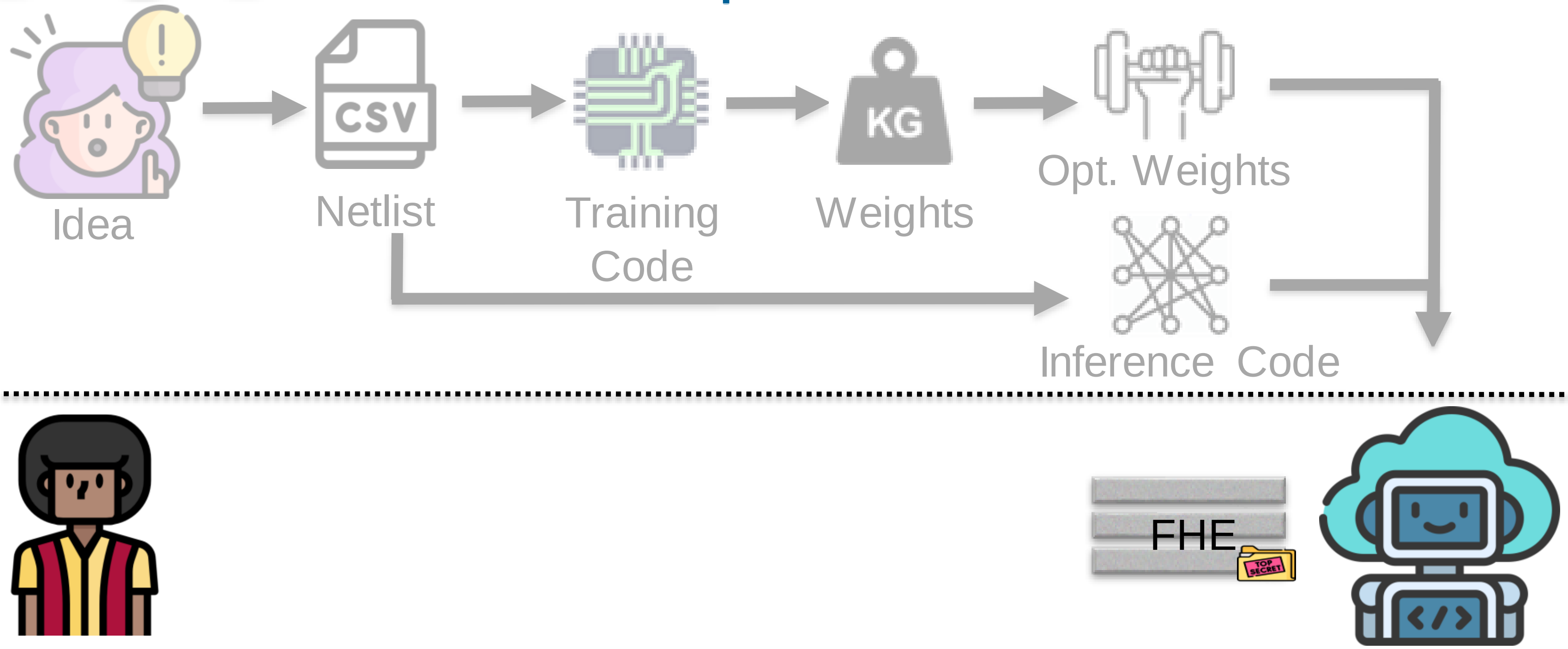
BYON

Example scenario

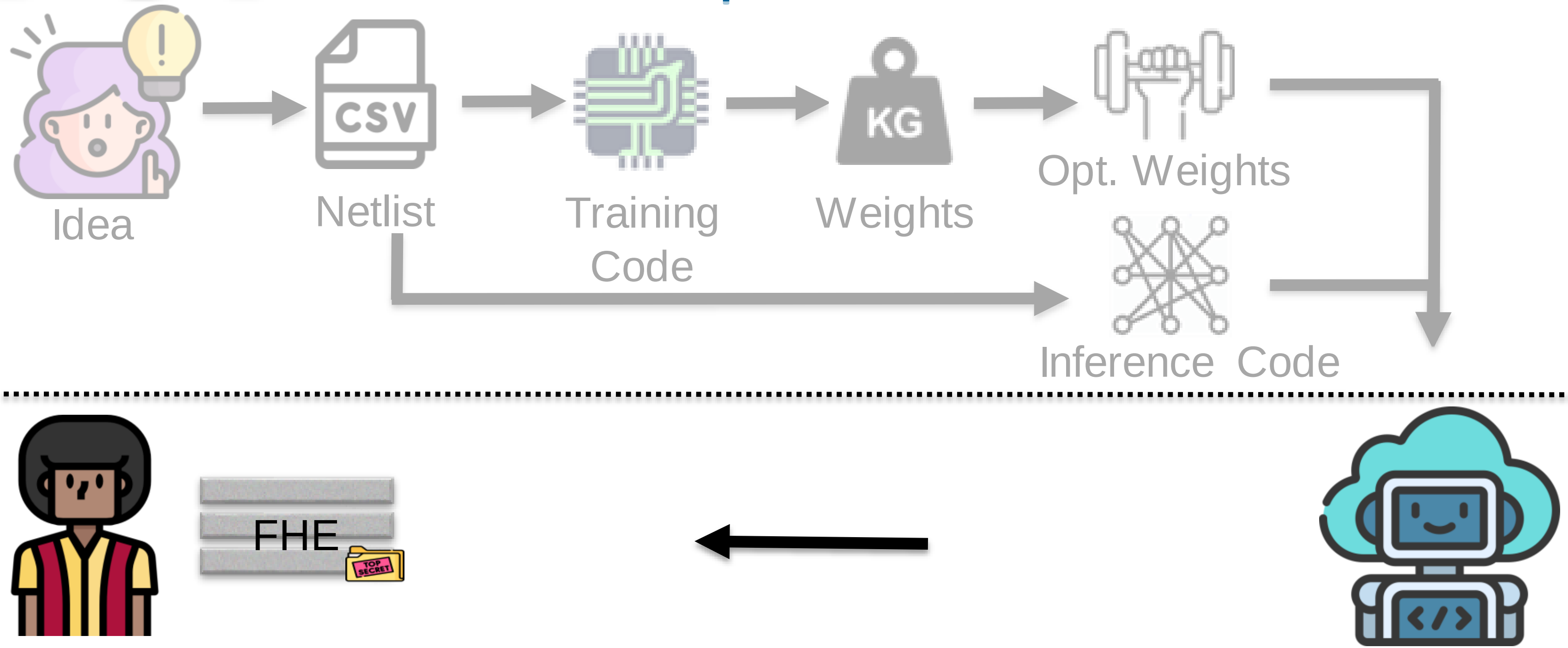


BYON

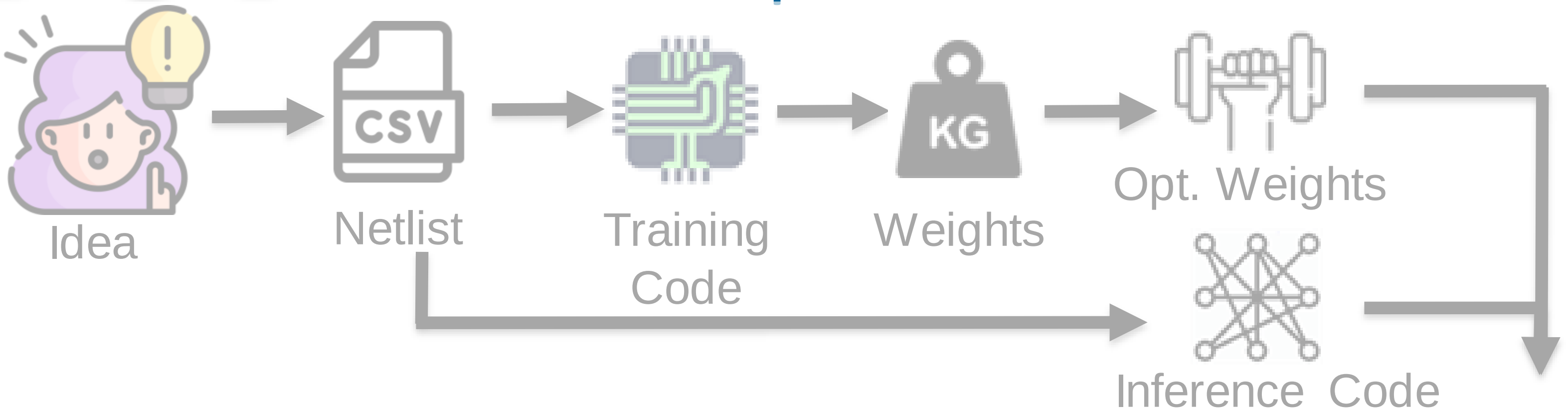
Example scenario

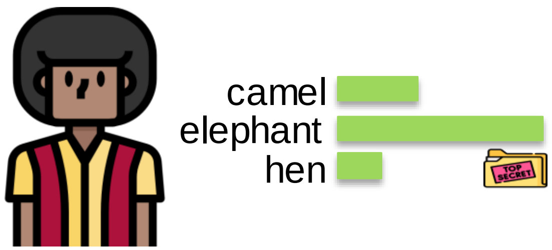


Example scenario



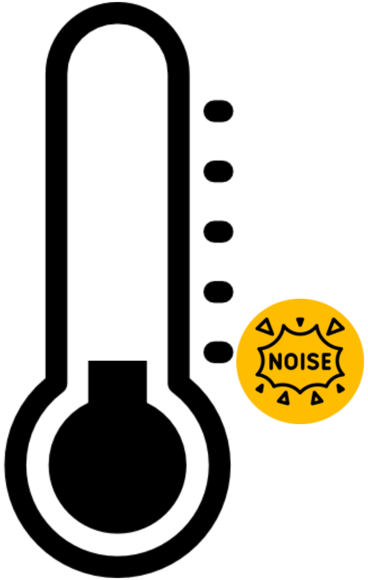
Example scenario





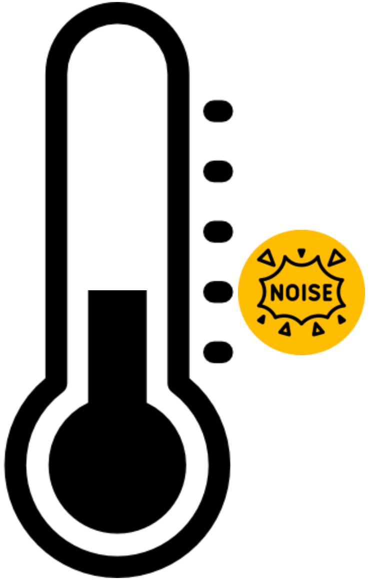
FHE 101

- Encrypt: Plaintexts become large polynomials
- Added noise guarantees security
- Noise growth bounds computation



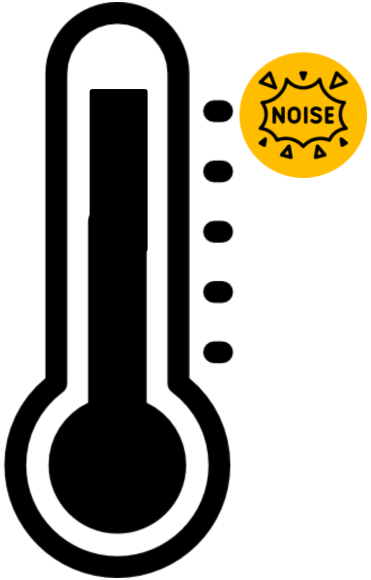
FHE 101

- Encrypt: Plaintexts become large polynomials
- Added noise guarantees security
- Noise growth bounds computation



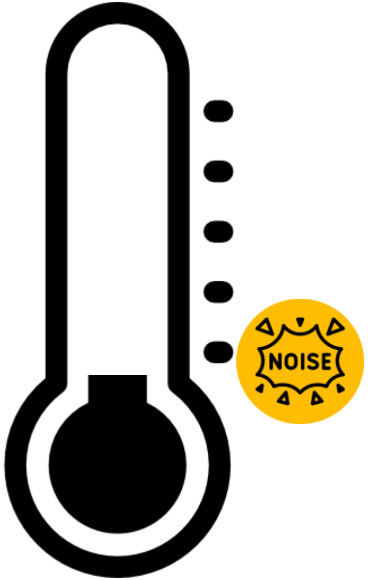
FHE 101

- Encrypt: Plaintexts become large polynomials
- Added noise guarantees security
- Noise growth bounds computation



FHE 101

- Encrypt: Plaintexts become large polynomials
- Added noise guarantees security
- Noise growth bounds computation
- Bootstrapping mitigates noise
    - Allows for *unbounded* arithmetic
    - High latency

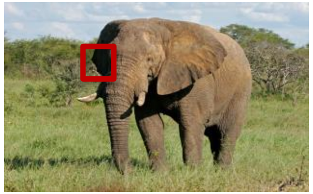


REDsec Bootstraps Only When Needed

- Noise auto-tuning
  - Noise grows predictably
  - Pinpoint bootstrapping locations on first inference
  - 32x improvement

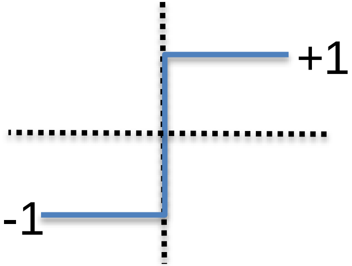
Discretized NNs 101

Convolution
Fully Connected



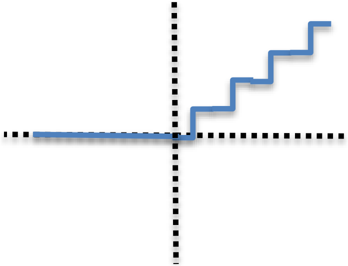
Multiply-Adds
Weights are {-1, 0, +1}

Sign Activation



y=(x<0)?-1:1

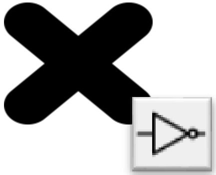
Discretized ReLU
Activation



y=max(0,x)

FHE-Friendly Operations

Convolution
Fully Connect



Efficient
Multiplication

Bridging
To Integer

Integer
Addition
Data Reuse

FHE-Friendly Operations

Convolution
Fully Connect

Efficient Multiplication

Bridging To Integer

Integer Addition Data Reuse

Sign Activation

Bias

Bridging To Binary

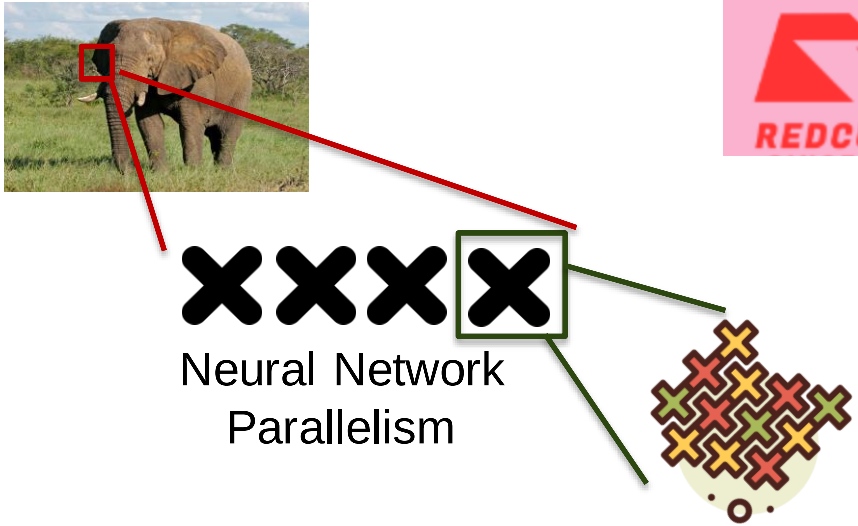
Sign Bit

FHE-Friendly Operations

Convolution Fully Connect	Sign Activation	ReLU Activation
Efficient Multiplication	Bias	Bias
Bridging To Integer	Bridging To Binary	Bridging to Binary
Integer Addition Data Reuse	Sign Bit	AND Sign Bit

Multi-GPU Acceleration on CUDA

Lots of parallelism!!

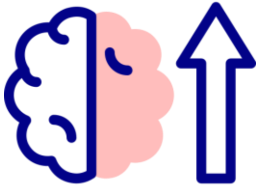


Neural Network Parallelism

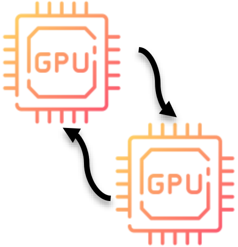
FHE Operation Parallelism

REDCUFHE

High Scalability!!



Dynamic Scheduler



Fewer Memory Transfers

Summary of REDsec Contributions

Efficient Multiplication

Bidirectional Bridging

Integer Addition Data Reuse

Binary Activations

BYON Framework

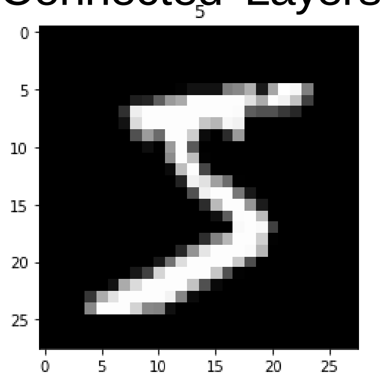
Noise Autotuning

GPU Acceleration

Scalability

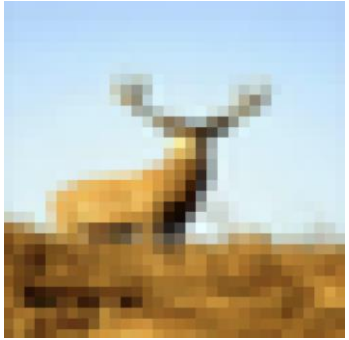
Experimental Evaluation

MNIST
1024 Neuron Fully Connected Layers



2.3 Million Multiply-Adds
3.6 seconds
3055x faster[1]

Cifar10
BinaryNet Architectures



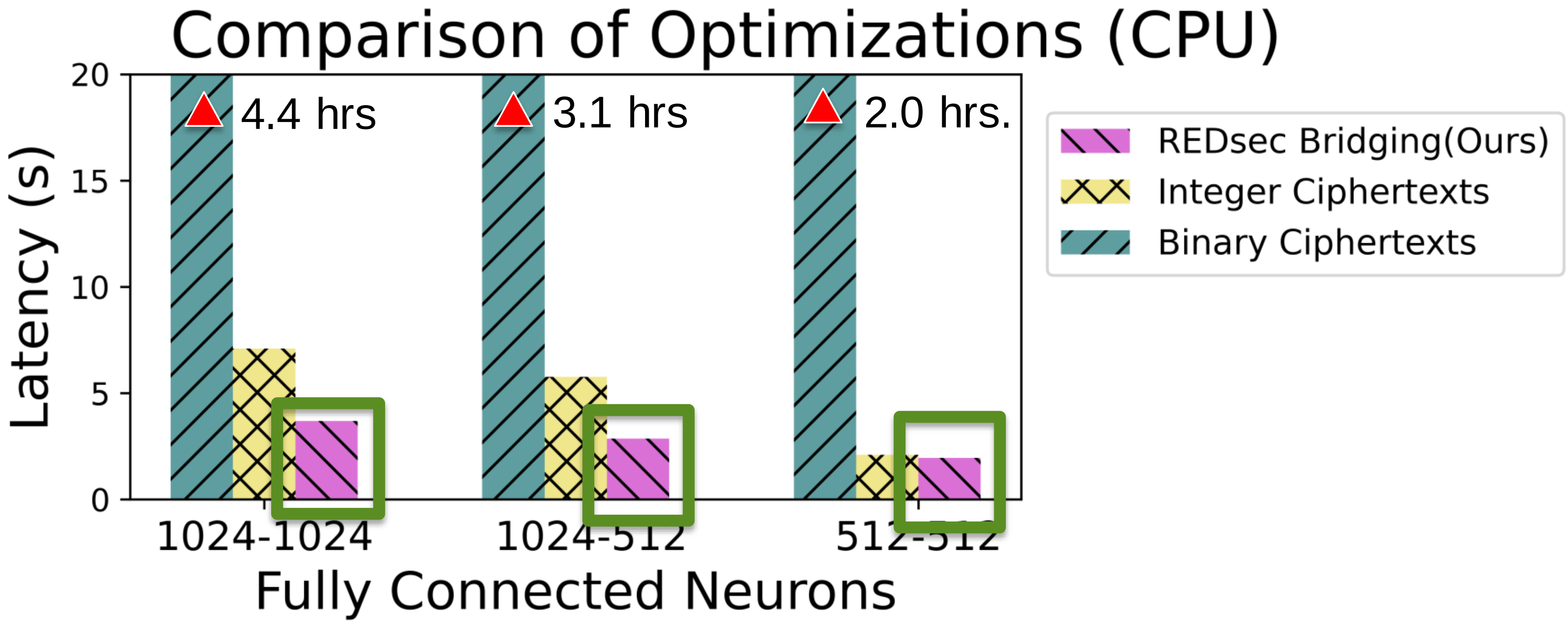
70 Million Multiply-Adds
3.8 minutes
11790x faster[1]

ImageNet
Binary AlexNet

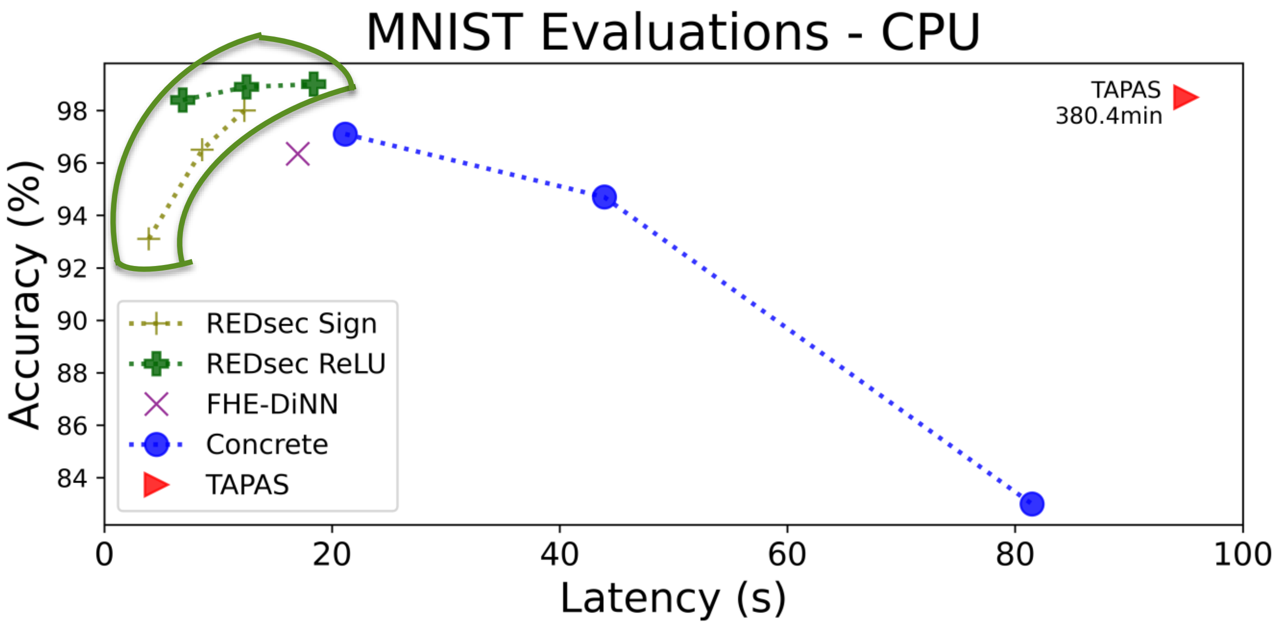
841 Million Multiply-Adds
1.6 hours
12166x faster[1]

[1] Neurips 2019

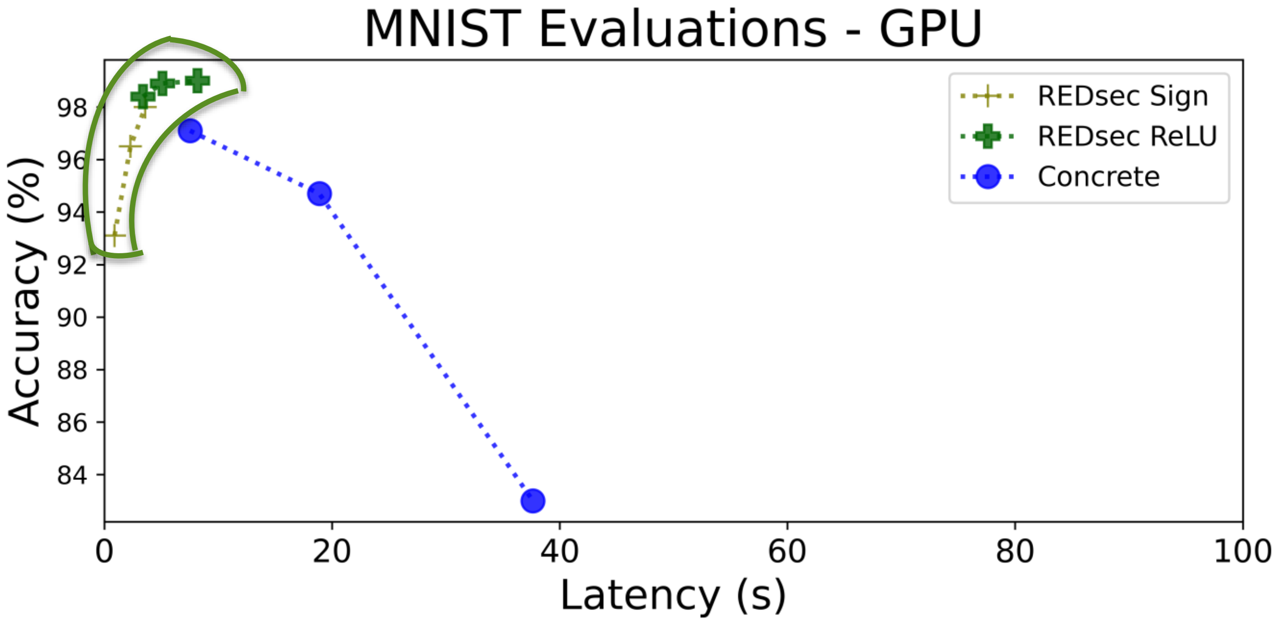
Impact of REDsec optimizations



CPU Comparisons to SoTA



GPU Comparisons to SoTA



Download REDsec today!

https://github.com/TrustworthyComputing/REDsec

folkerts@udel.edu