MATHEMATICAL FRONTIERS

The National Academies of SCIENCES ENGINEERING MEDICINE

nas.edu/MathFrontiers

Board on Mathematical Sciences & Analytics

MATHEMATICAL FRONTIERS
2019 Monthly Webinar Series, 2-3pm ET

February 12: *Machine Learning for Materials Science*

March 12: *Mathematics of Privacy*

April 9: *Mathematics of Gravitational Waves*

May 14: *Algebraic Geometry*

June 11: *Mathematics of Transportation*

July 9: *Cryptography & Cybersecurity*

August 13: *Machine Learning in Medicine*

September 10: *Logic and Foundations*

October 8: *Mathematics of Quantum Physics*

November 12: *Quantum Encryption*

December 10: *Machine Learning for Text*

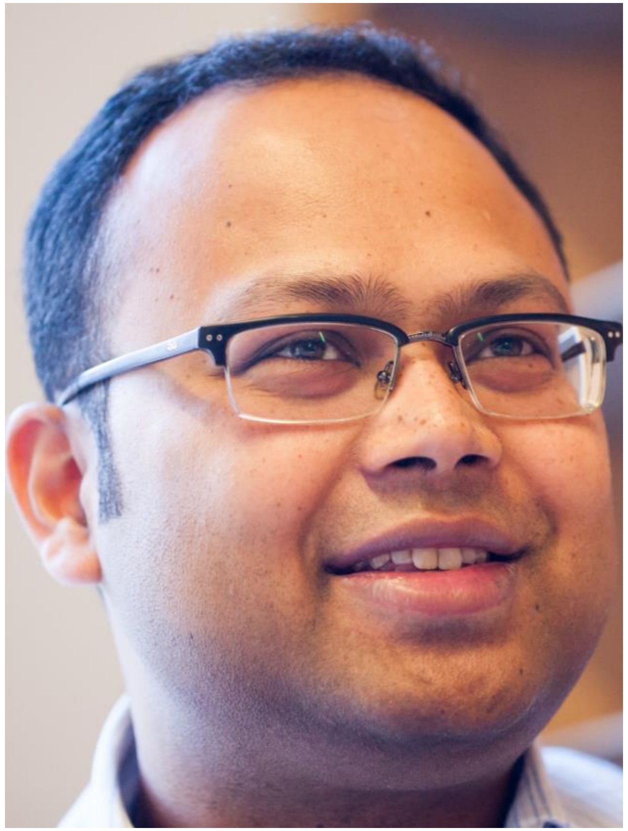
Made possible by support for BMSA from the
National Science Foundation
Division of Mathematical Sciences
and the
Department of Energy
Advanced Scientific Computing Research

View webinar videos and learn more about BMSA at www.nas.edu/MathFrontiers

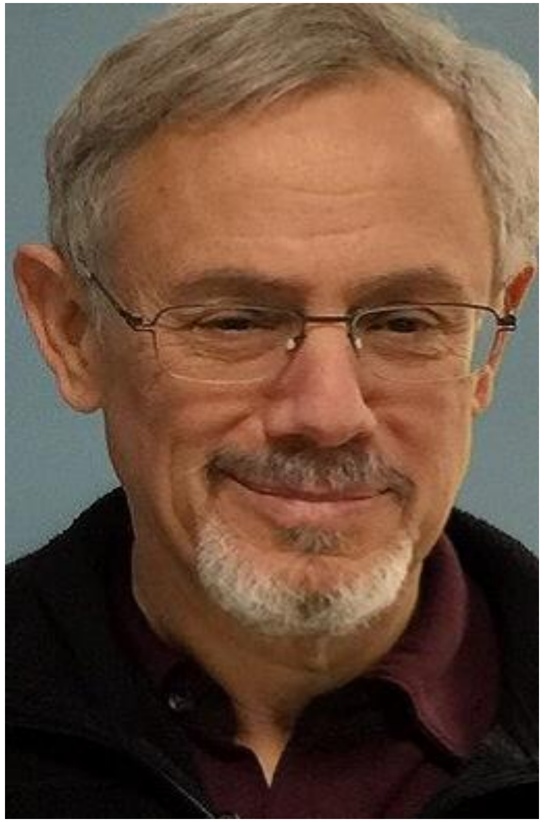
MATHEMATICAL FRONTIERS

Cryptography and Cybersecurity

Kristin Lauter,
Microsoft Research



Suman Jana,
Columbia University



Mark Green,
UCLA (moderator)

Kristin Lauter,
Microsoft Research

Principle Researcher and Research Manager for Cryptography

Private AI: Machine Learning on Encrypted Data

Private AI:
Machine Learning on Encrypted Data

Kristin Lauter

Partner Research Manager,
Principal Researcher
Cryptography Research

Microsoft Research
SEAL Team: sealcrypto.org

July 9, 2019

Microsoft

Privacy problem with AI?

- Artificial Intelligence: uses (ML) machine learning algorithms to make useful predictions
  - Input: your data
  - Output: some recommendation, decision, or classification
- **Privacy problem**: you have to input **your data** in order to get the valuable prediction!
- Typical AI services are hosted in the cloud, run by a “smart agent” (e.g. Cortana, Siri, Alexa)

New mathematical tool

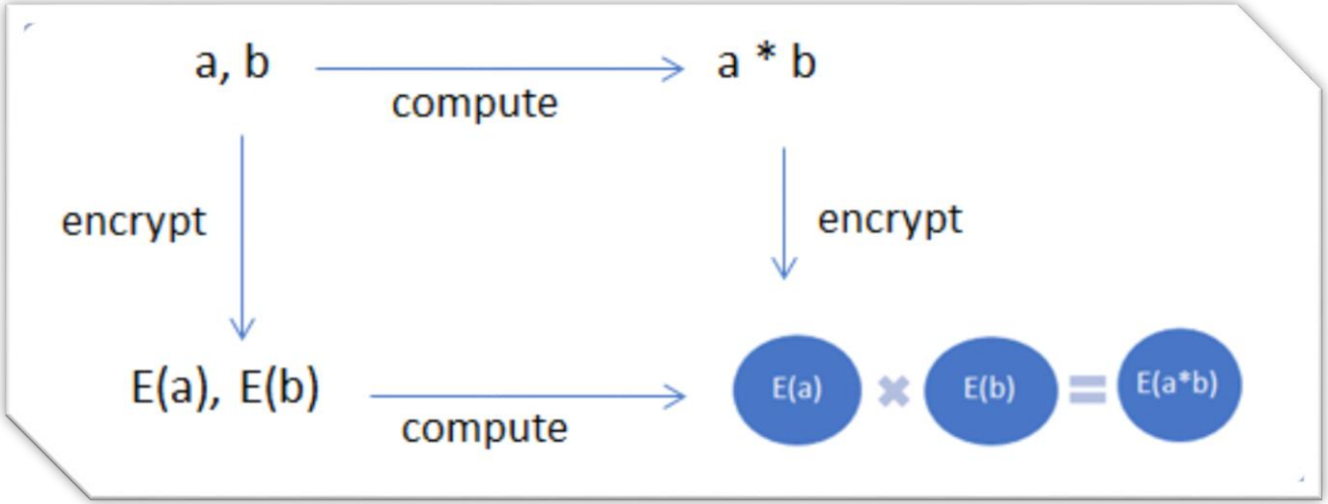
- Protect privacy and security of your data through encryption

- Need encryption which you can compute on:

    - **Homomorphic Encryption!**

Homomorphic Encryption (HE)

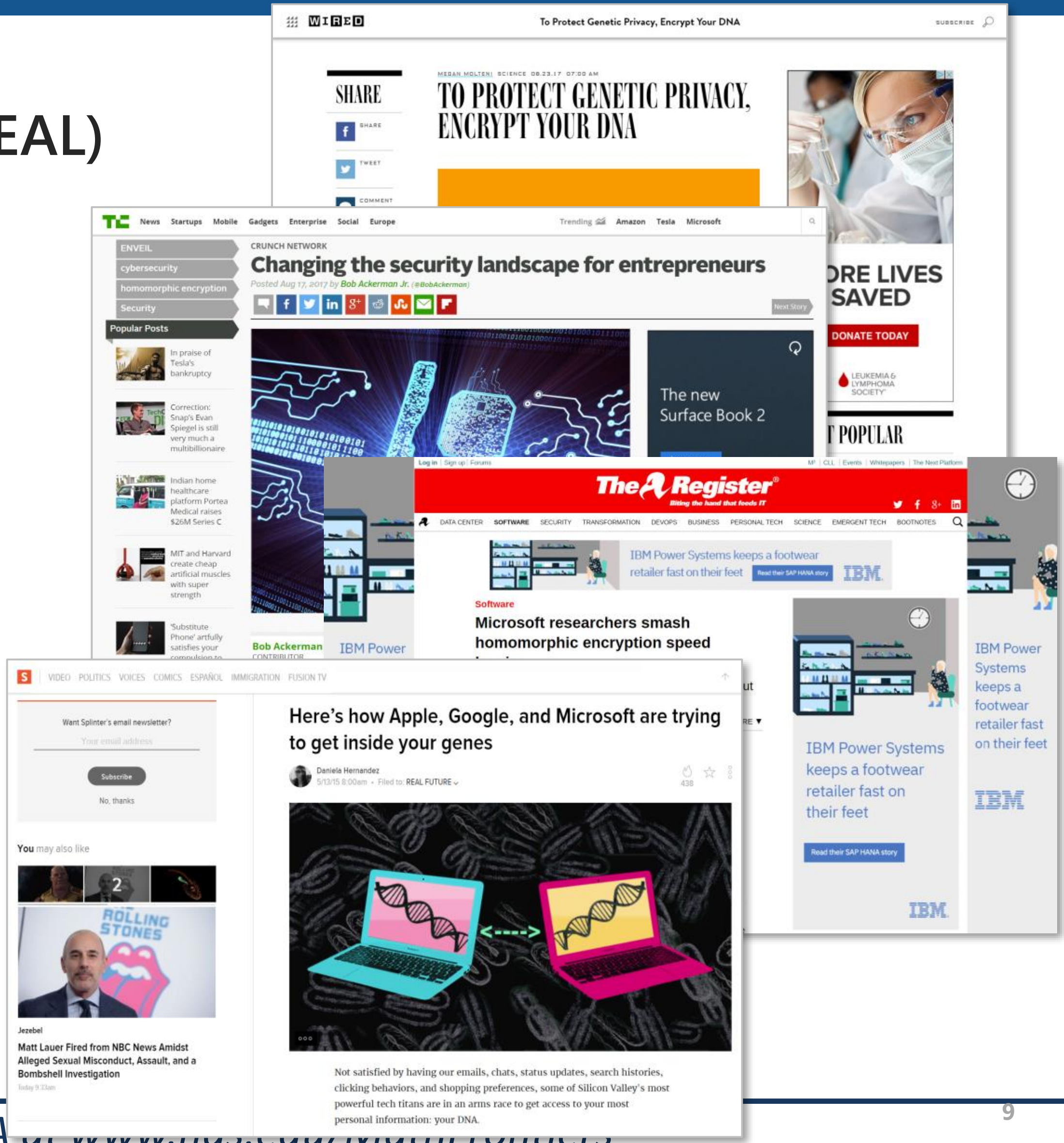
- Computation on encrypted data without decrypting it!
- 2009: First solution, considered impractical
- 2011: Surprise breakthrough at Microsoft Research
- Practical encoding: 4 orders of magnitude speed-up
- 2016: **CryptoNets** evaluates neural net predictions on encrypted data



Microsoft SEAL

- **Simple Encrypted Arithmetic Library (Microsoft SEAL)**
    - Public release by Microsoft Research in 2015
    - Widely adopted by teams worldwide
    - OSS release of Microsoft SEAL 2018
- **Standardization of HE:**
    - November 2018

http://sealcrypto.org



Standardization: HomomorphicEncryption.org

- MSR launched in 2017
- Three workshops: Microsoft, MIT, U Toronto
- HES 1.0 Standard, November 2018
- Royal Society (PET) Report 2019
- Applications in regulated industries require standardization
- Standardization creates trust
- Open standards highly preferred in cryptography

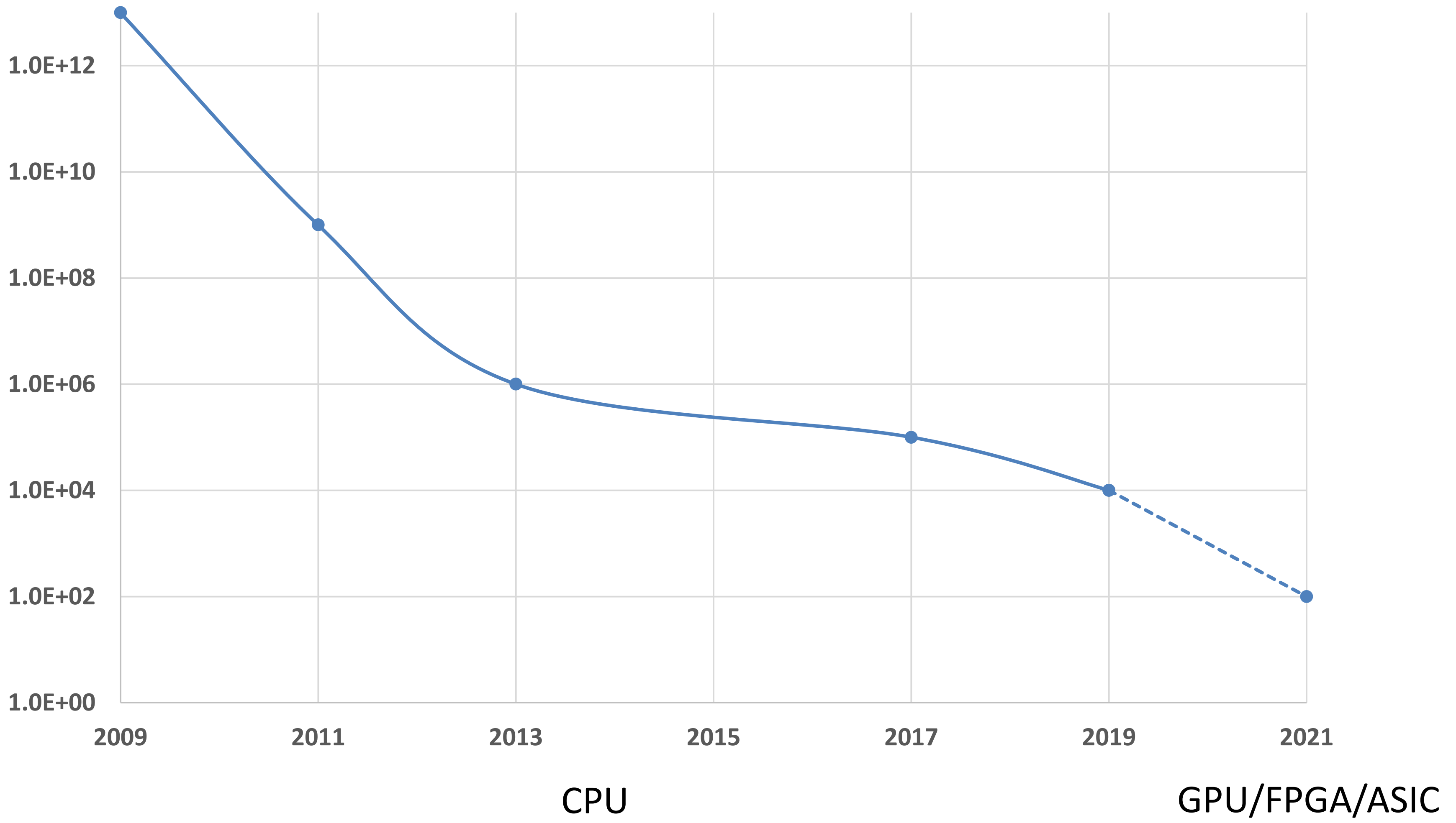
Mathematics of Homomorphic Encryption

- **New hard problems** proposed (2004-2013)
  - ❑ Small Principal Ideal Problem, Approximate GCD,
  - ❑ Learning With Errors (LWE), Ring-Learning With Errors (RLWE)
  - ❑ Related to well-known hard lattice problems:
- **Lattice-based Cryptography:**
  - ❑ Proposed by Hoffstein, Pipher, and Silverman in 1996 (NTRU), Aijtai-Dwork
  - ❑ Compare to other public key systems: RSA (1975), ECC (1985), Pairings (2000)
- **Hard Lattice Problems:**
  - ❑ Approximate Shortest Vector Problem (SVP), Bounded Distance Decoding
  - ❑ 30 year history of Lattice-basis reduction (LLL, BKZ, BKZ 2.0, FpLLL, sieving, challenges)
- **Security:**
  - ❑ Best attacks take exponential time
  - ❑ Secure against quantum attacks (so far…)

High-level Idea

- **Encryption:**
  - Encryption adds noise to a “secret” inner product
  - Decryption subtracts the secret inner product and the noise becomes easy to cancel
- **Hard Problem:**
  - Hard problem is to “decode” noisy vectors
  - If you have a short basis, it is easy to decompose vectors
- **Homomorphic property:**
  - Lattice vectors → coefficients of polynomials
  - Polynomials can be added and multiplied

Performance overhead improvement over Time (log scale!)



Microsoft SEAL 3.2 Performance Numbers

Machine : Intel Core i7-8700k @ 3.70 GHz; Single Thread

PolyModulus Degree	Encrypt	Encrypt Amort.	Decrypt	Decrypt Amort.	Add	Multiply	Multiply Amort.
4096	2683 us	1.31 us	1776 us	17 us	0.008 us	517 us	**0.252 us**
8192	7114 us	1.7 us	6091 us	72 us	0.018 us	2746 us	**0.67 us**
16384	21000 us	2.5 us	25000 us	361 us	0.044 us	17000 us	**2.066 us**
32768	69000 us	4.24 us	118000 us	1877 us	0.114 us	105000 us	**6.410 us**

Private AI Demos History

- 2014: Heart attack risk, personal health data ~ 1 second
- 2015: CryptoNets demo showing neural net prediction: MNIST data set ~80 seconds
- 2016: Genomics predicting flowering time from 200K SNPs ~ 1 second
- 2016: Pneumonia mortality risk: intelligible models ~ 8 seconds for 4,000 predictions

- 2018: Twitter sentiment analysis (150K text features) ~ less than a second
- 2018: cat/dog image classification ~ less than a second

- 2019: Asure Run (Private Fitness App)
- 2019: Chest Xray diagnostics
- 2019: Secure Weather prediction

Demos

Demo 1 : AsureRun (Private Fitness App)

01

Private Storage and Analytics

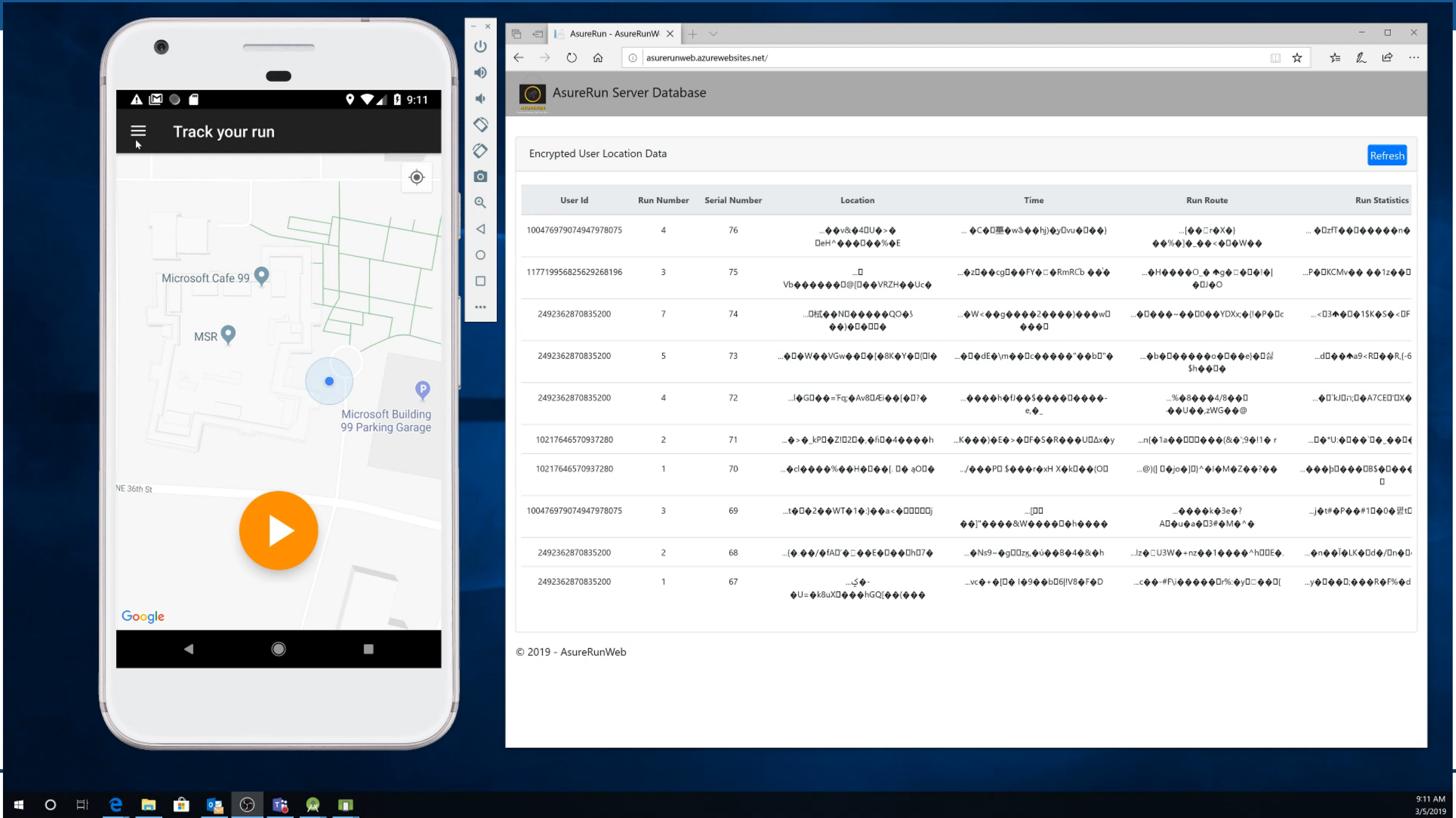
02

Private AI Prediction Services

03

Hosted Private Training

Scenario 1: AsureRun



Demo 2 : Chest X-Ray (disease prediction)

01

Private Storage and Analytics

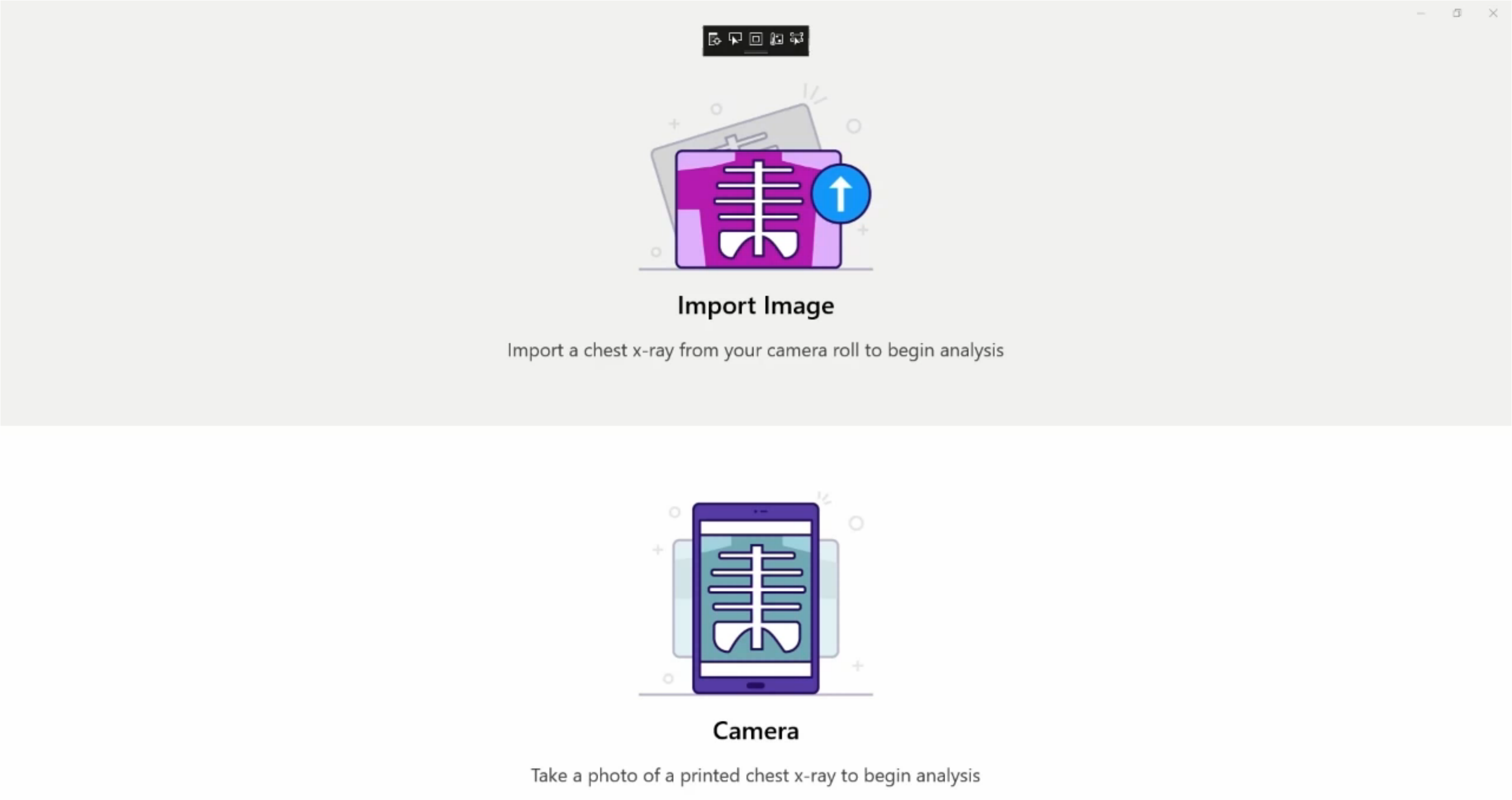
02

Private AI Prediction Services

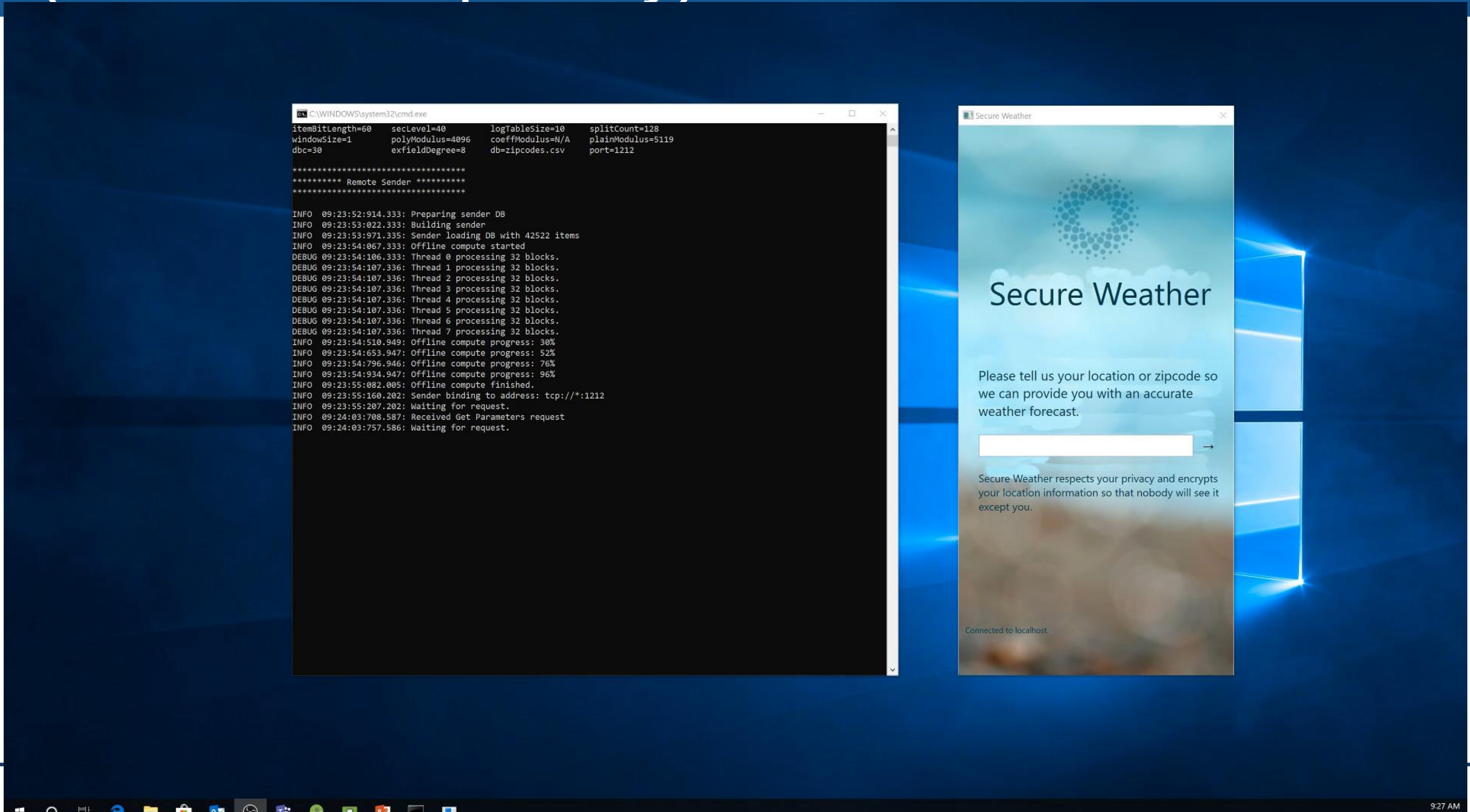
03

Hosted Private Training

Chest X-ray


Import Image
Import a chest x-ray from your camera roll to begin analysis
Camera
Take a photo of a printed chest x-ray to begin analysis

Scenario 3: Weather Prediction (with location privacy)



Resources

Microsoft Research project:

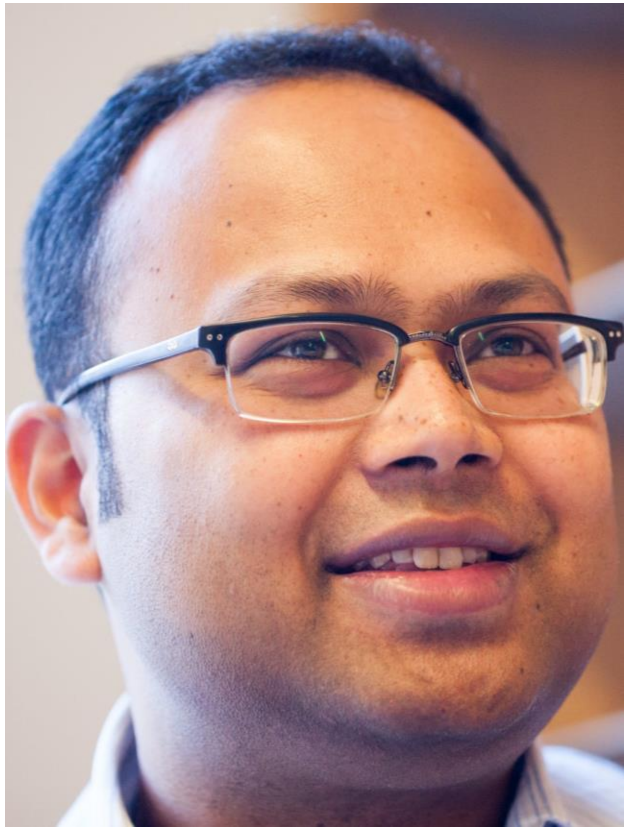
https://www.microsoft.com/en-us/research/project/homomorphic-encryption/

Standardization community:

http://homomorphicencryption.org/

SEAL code to download:

https://www.microsoft.com/en-us/research/project/microsoft-seal/



Suman Jana,
Columbia University

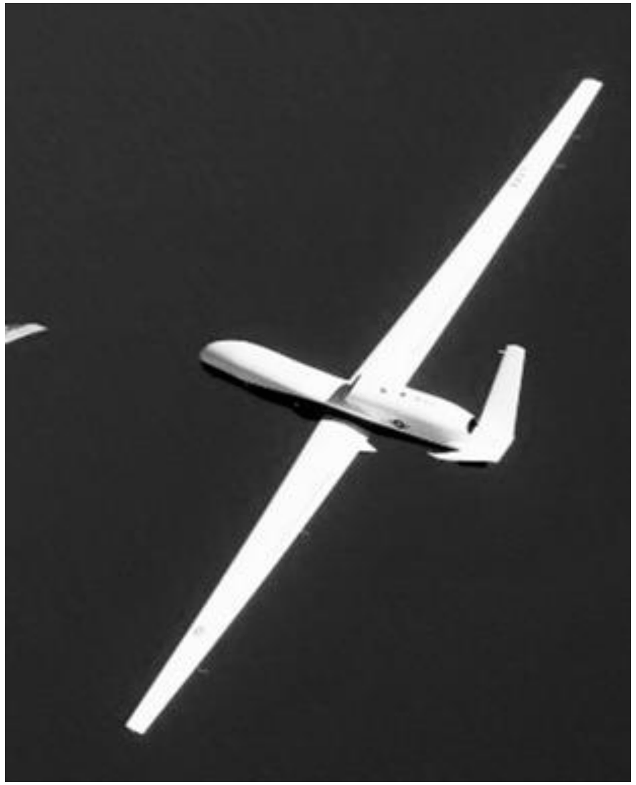
Assistant Professor of Computer Science

Formal verification of Neural Networks

Machine Learning Systems in Security-critical Domains

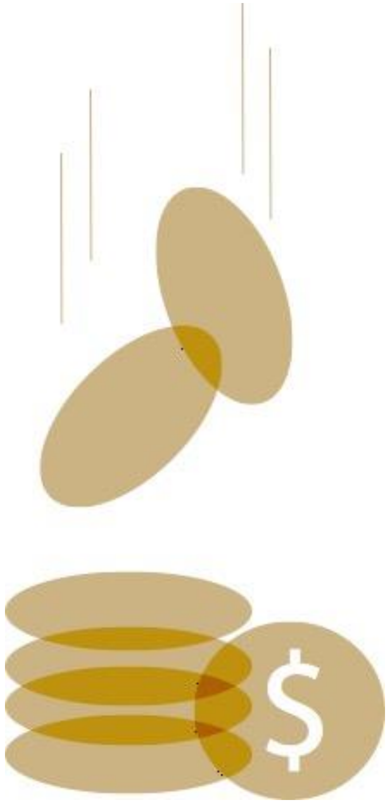
Autonomous Vehicles

(Tesla, Google, ...)



Unmanned Aircraft Systems

(Navy MQ-4C Triton)



Huge potential benefits

Machine Learning: Safe and Secure?

Tesla Autopilot fatal crash, June 2017

Tesla Autopilot fatal crash, March 2018

Uber fatal crash, June 2018

Not robust or reliable

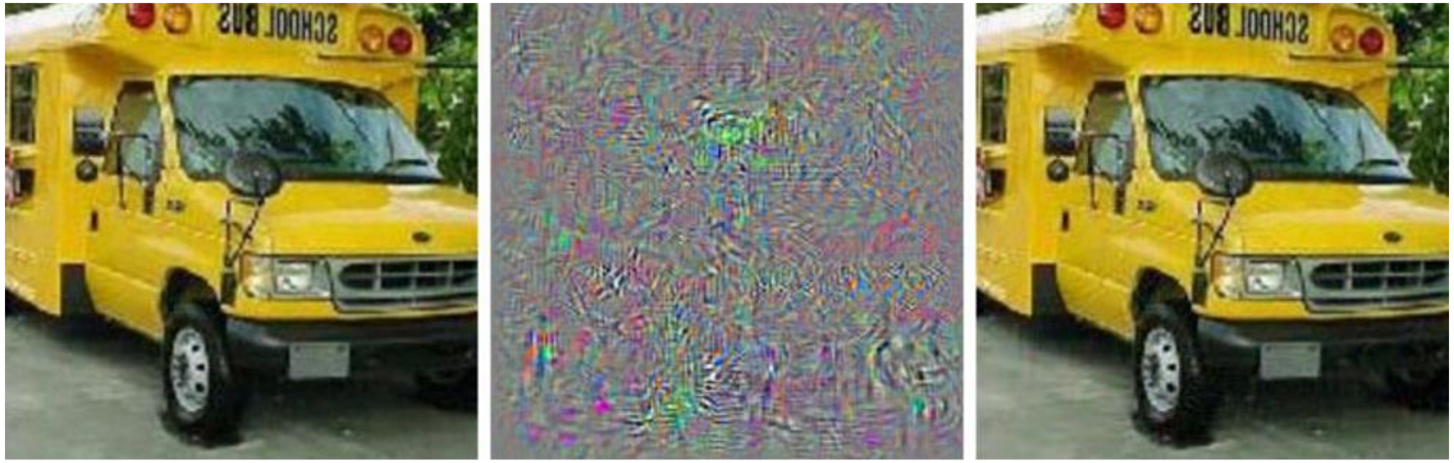
Danger

Adversarial Examples

Adversarial examples:
- Minor perturbations will lead to misclassifications with high confidence

An inherent weakness of neural networks

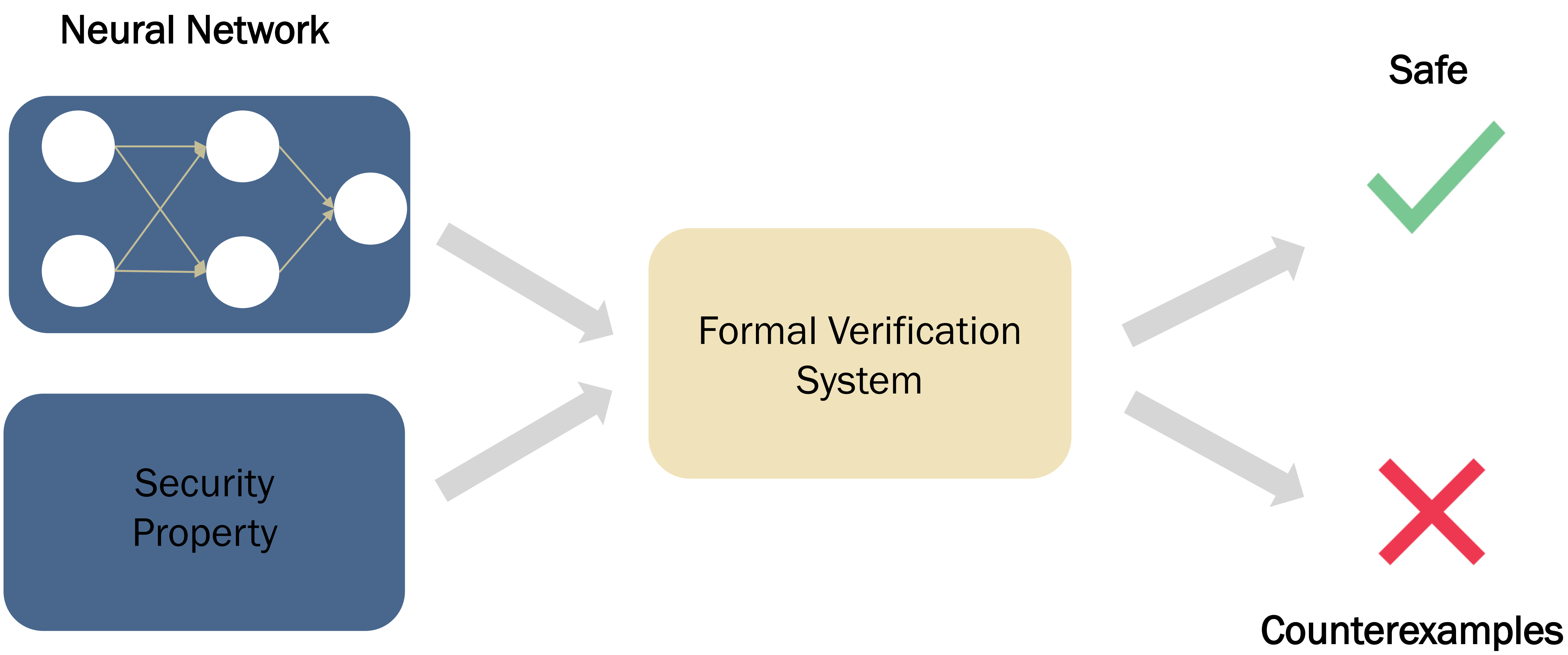
No scalable technique to prove non-existence of adversarial examples



Bus | Adversarial Noise | Ostrich

Formal analysis is urgently needed

Formal Verification of Neural Networks



Security Properties

Given a neural network

Given input ranges

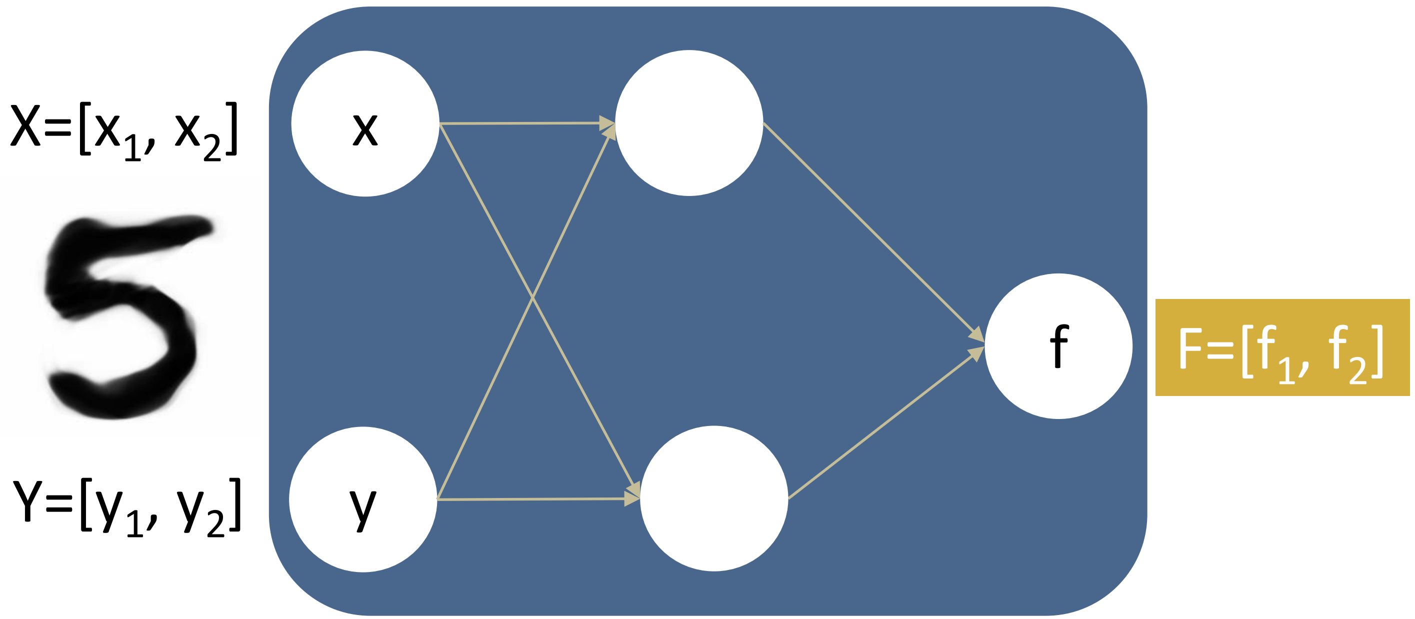
- p-norm for images
- Customized ranges

Check output range F

✓ Safe: F satisfies property P

✗ Unsafe: counterexample found

Input layer Hidden layers Output layer



Hard to analyze

Hardness of Formally Bounding Output Ranges

NN without activation functions

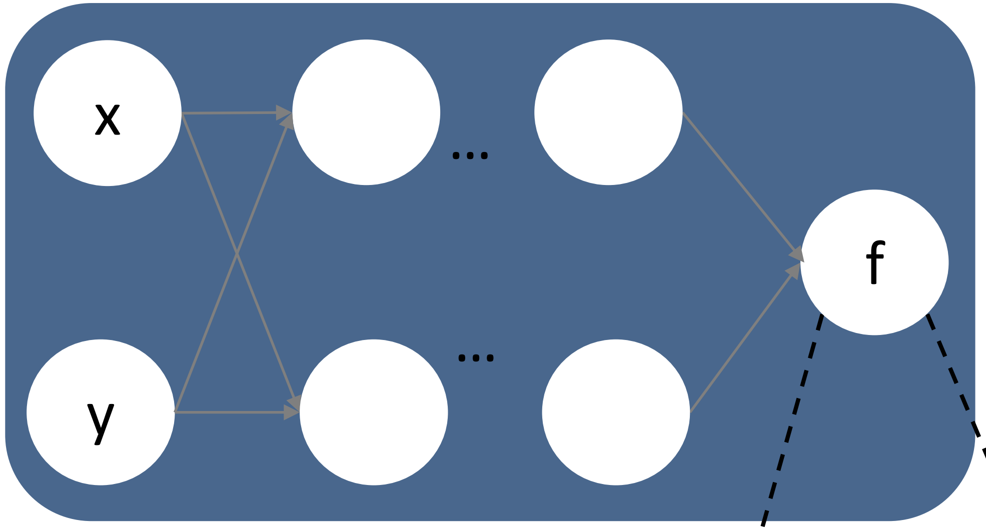
- Linear, easy to analyze
- Not very useful in practice

ReLU

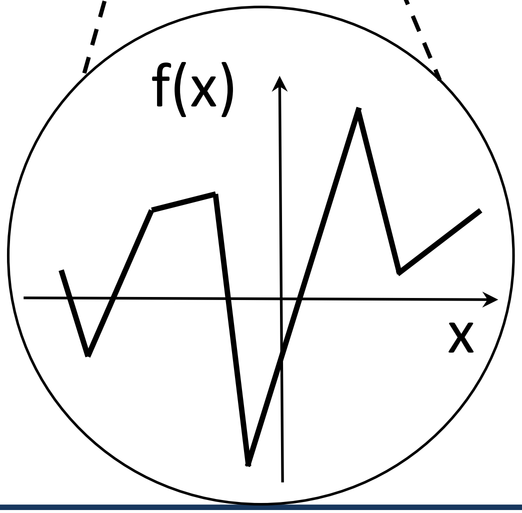
- ReLU(x)=max(x,0)
- Two linear pieces

ReLU-based NN

- Outputs are determined by combination of ReLU pieces
- Linear pieces exponentially increase as NN gets larger



d^n linear pieces



Very expensive to track all linear pieces

Difference with formal verification of traditional software

Complete specification is not feasible
- Mostly local security properties with respect to an existing dataset (e.g., adversarial perturbations to images in a dataset should not change classifier output)

ML verification can provide verified lower bounds on robustness of a ML model against different attackers
- All perturbations bounded by L_inf norm
- All rotation angles within some bounded range

Generalizability
- Will verified robustness on a test dataset hold for new datasets?
- Machine learning classifiers are already making this assumption for regular accuracy
- Verification is simply leveraging generalizability to provide strong robustness guarantees

Existing Approaches: Customized Solvers

Extend SMT/linear/MILP solvers

- ReluPlex by Katz et al. [CAV'17]
- Sherlock by Dutta et al. [NFMS'18]

Limitations

- High overhead even for simple properties (>hours or days)
- Hard to scale for large networks (<=1000s of ReLU nodes)

Existing Approaches: Relaxation

Over-approximate NN output ranges

- Over-approximate with abstract domain
- Relax to convex optimization problems

Several existing works

- Kolter et al. [ICML'18]
- Gehr et al. [Oakland'18]
- Dvijotham et al. [arXiv:1803:06567]

Drawbacks:

- High false positive rates
- Inefficient at finding concrete violations

We want the best of both worlds!

Efficiently compute sound (i.e., over-approximated) bounds on NN outputs

Iteratively refine the bounds by spending increasingly more computation power

- Ensure that bound soundness is maintained at each step
- Perform cheap, targeted search for counterexamples in each iteration

Support fine-grained trade-off between over-approximation errors and computational effort

Interval Analysis

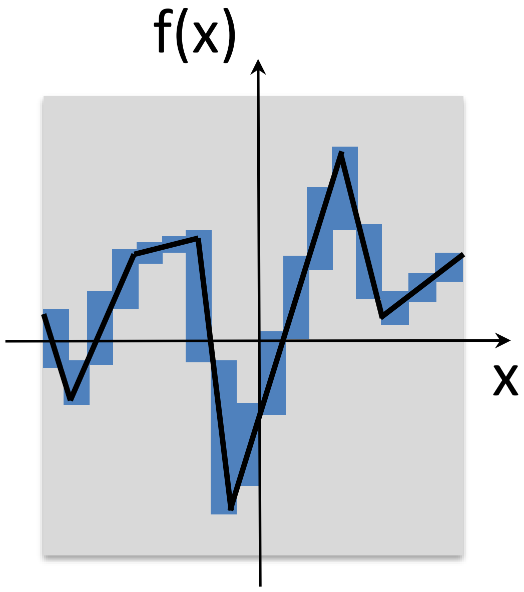
Sound (always over-approximate output)

Fast bound propagation

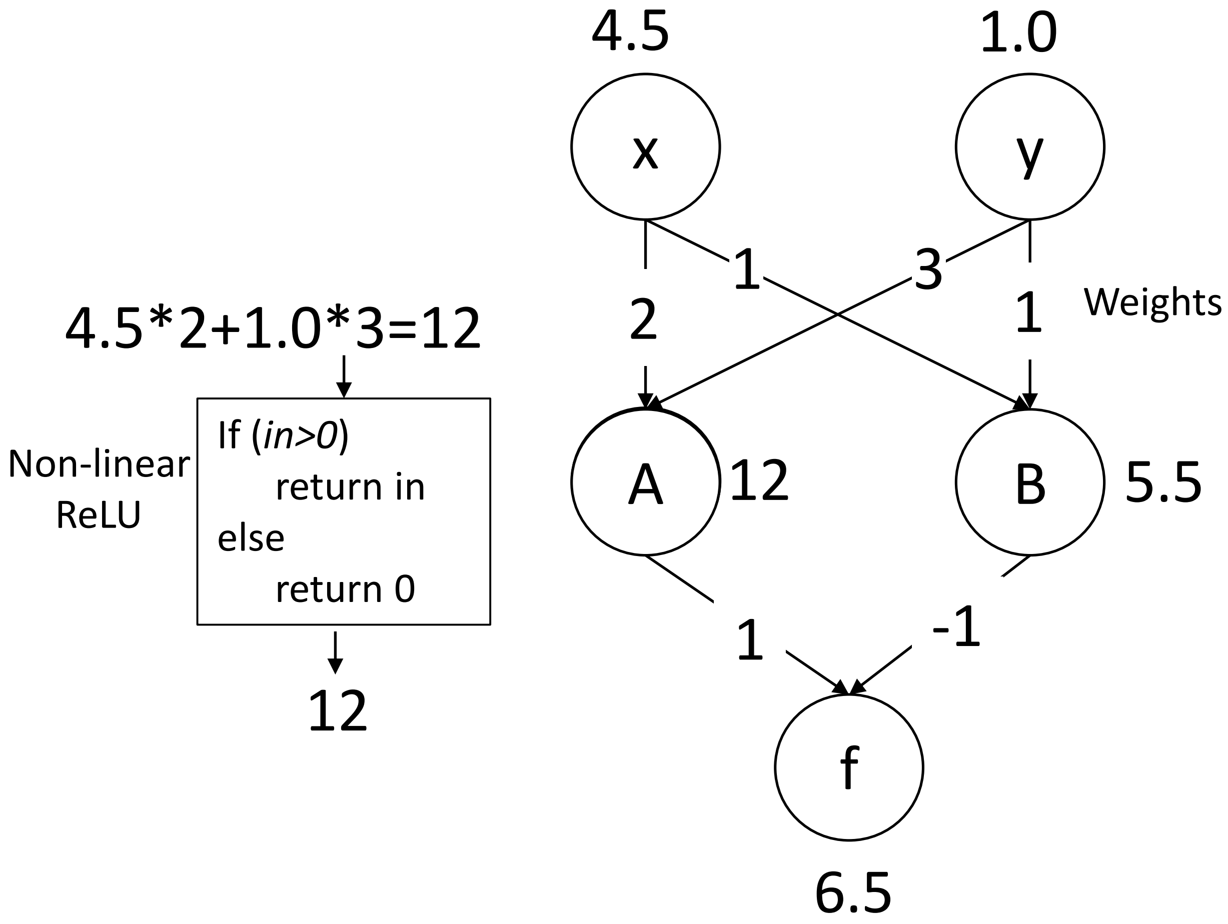
- Interval versions of common NN operations (addition and multiplication) are efficient

Easy and efficient refinement through bisection

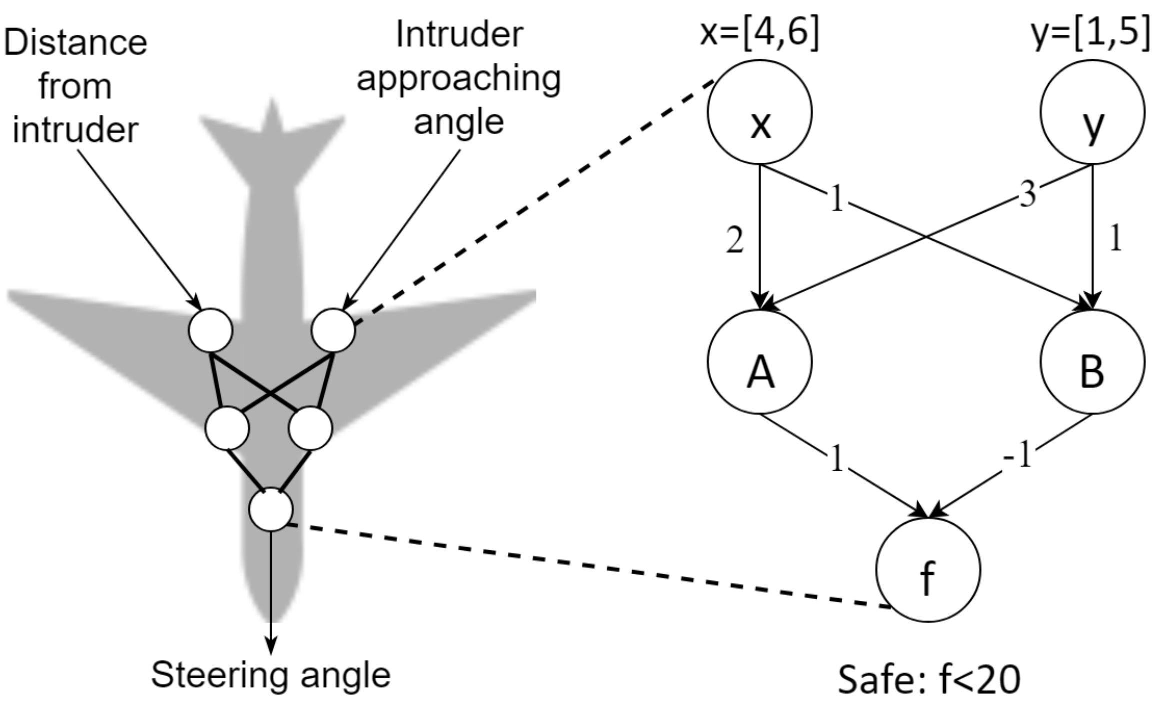
- Can approximate NN outputs up to any desired precision
- Amenable to highly parallelizable Branch and bound techniques

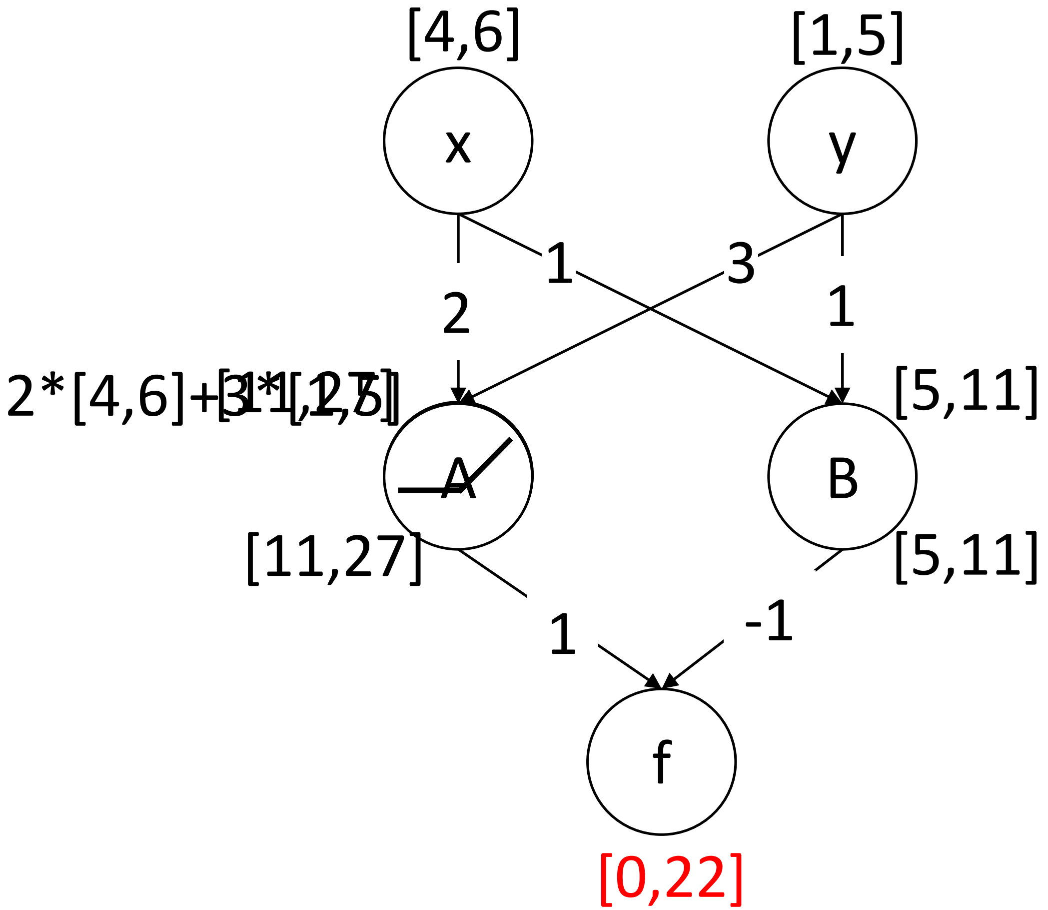


Neural network basics (inference)



Sample of Naive Interval Analysis





True security violation?

Security property:

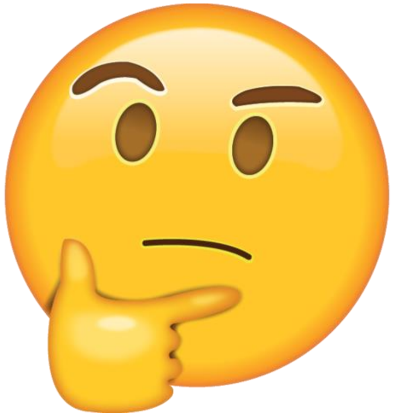
input ranges: x=[4,6], y=[1,5]

output: f<20

Ignoring Dependencies Cause Overestimation!

$x = [-1,1]$

$f = x - x = ?$



Naive interval:

$f = [-1,1] - [-1,1] = [-2,2]$

True bound:

$f = x - x = [0,0]$

Dependency matters

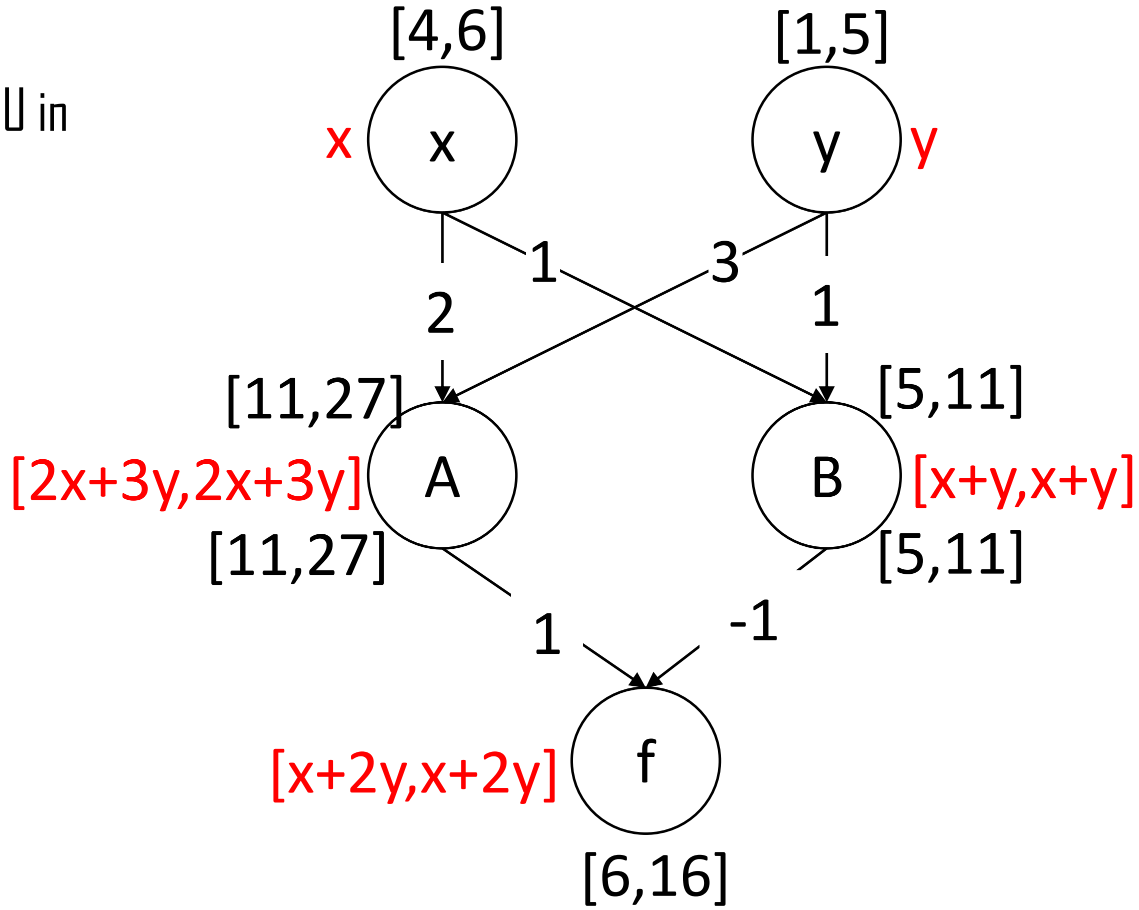
Symbolic Interval Analysis

Symbolic interval [Eq_{low}, Eq_{up}]

- Symbolic linear upper and lower bound equations for each ReLU in terms of inputs

Tighter output approximation

- Track input dependencies



f is always less than 20

Safe!

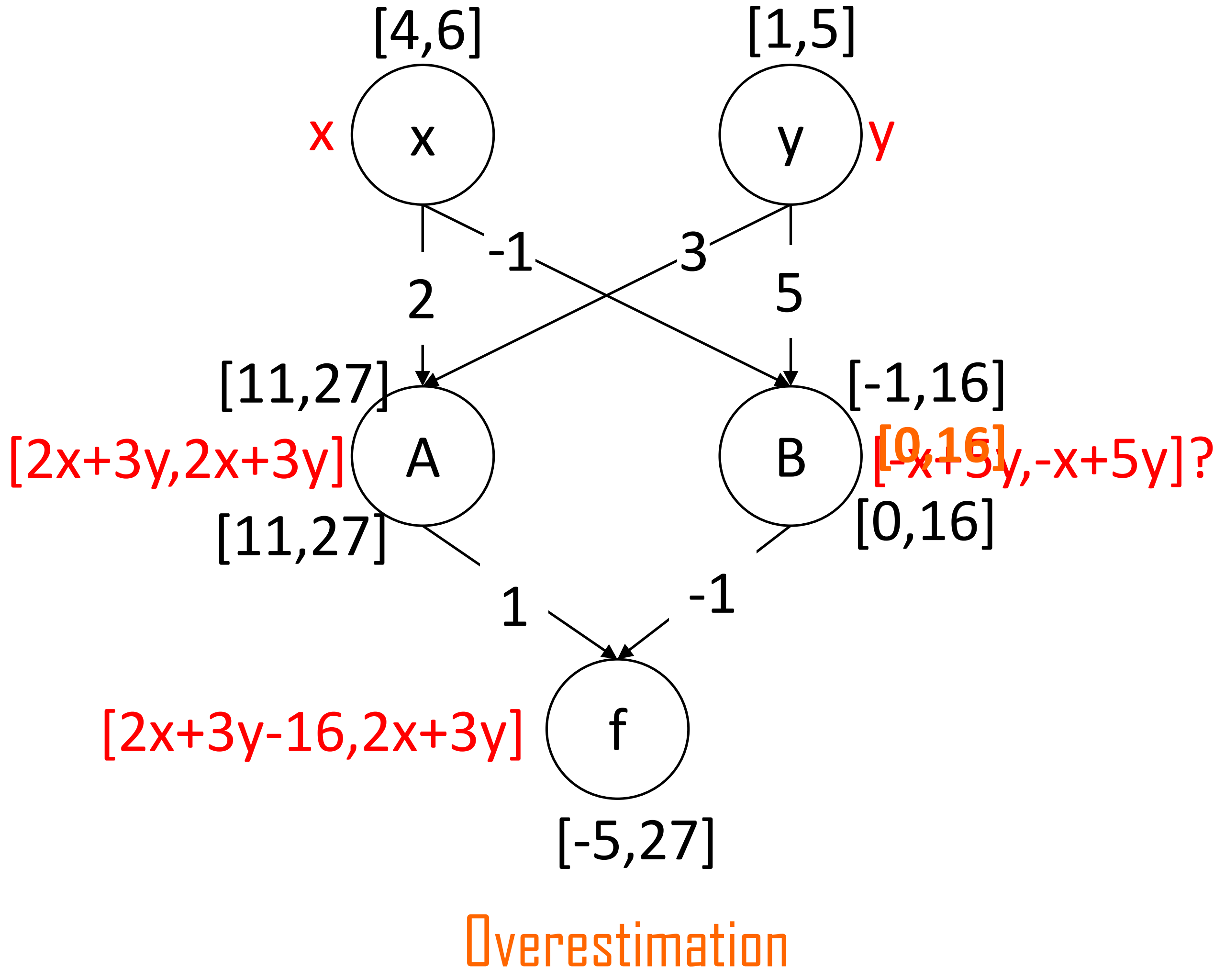
Security property:

input ranges: x=[4,6], y=[1,5]

output: f<20

Symbolic Interval Analysis is Not Perfect

Concretization
- Only when nonlinearity occurs
- Overestimation
- May have many false positives for large NNs



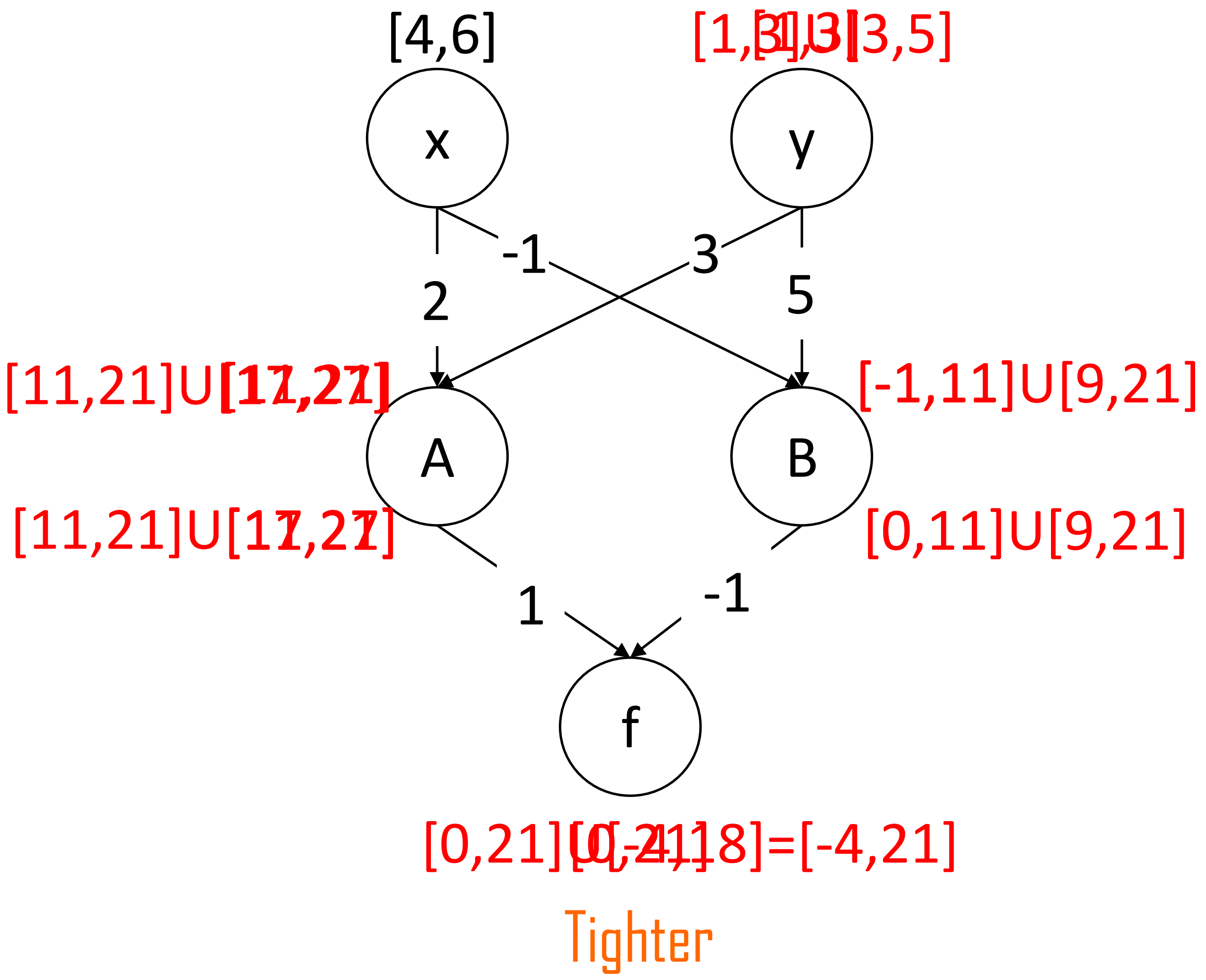
Security property:
input ranges: x=[4,6], y=[1,5]
output: f<20

Iterative Interval Refinement with One Bisection

Iterative refinement

- Bisect input ranges
- Compute union of output ranges
- Iteratively refine

Overestimation error decreases as input width becomes smaller



Security property:
input ranges: x=[4,6], y=[1,5]
output: f<20

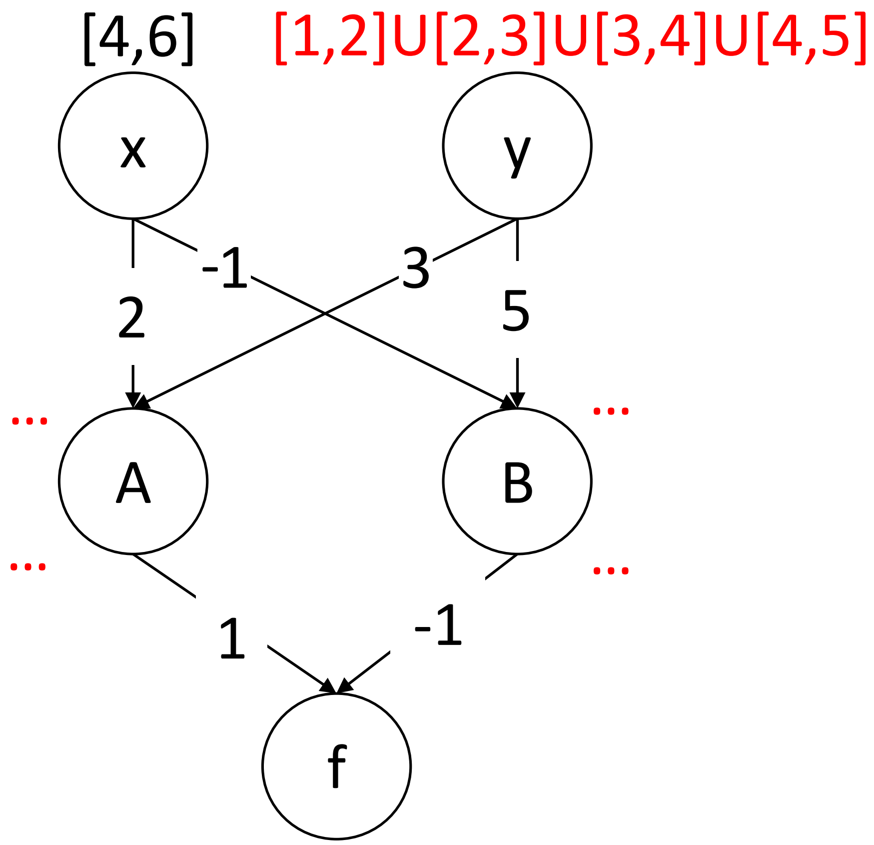
Iterative Interval Refinement with Two Bisections

Iterative refinement

- Bisect input ranges
- Compute union of output ranges
- Iteratively refine

Overestimation error decreases as input width becomes smaller

Security property:
input ranges: x=[4,6], y=[1,5]
output: f<20



[5,18]U[3,17]U[1,15]U[0,19]=[3,19]

f is always less than 20

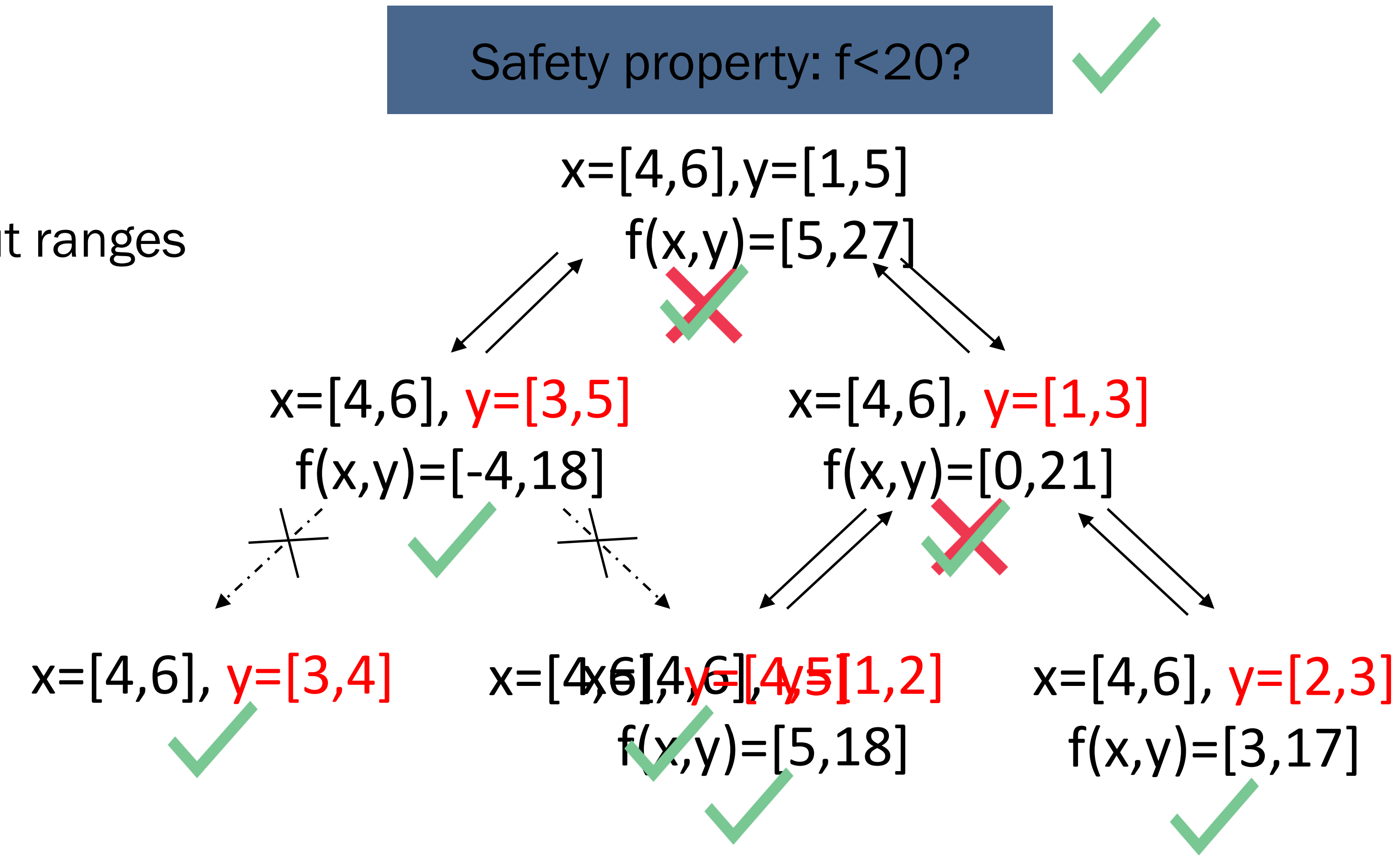
Safe!

Benefits of Iterative Interval Refinement

Highly uneven distribution
- No need to split safe subtrees
- Flexibly refine output based on the property

Iteratively narrow down violating input ranges
- Easily locate counterexamples

Highly parallelizable
- Subtrees are independent



ACAS Xu Models

Airborne collision avoidance system

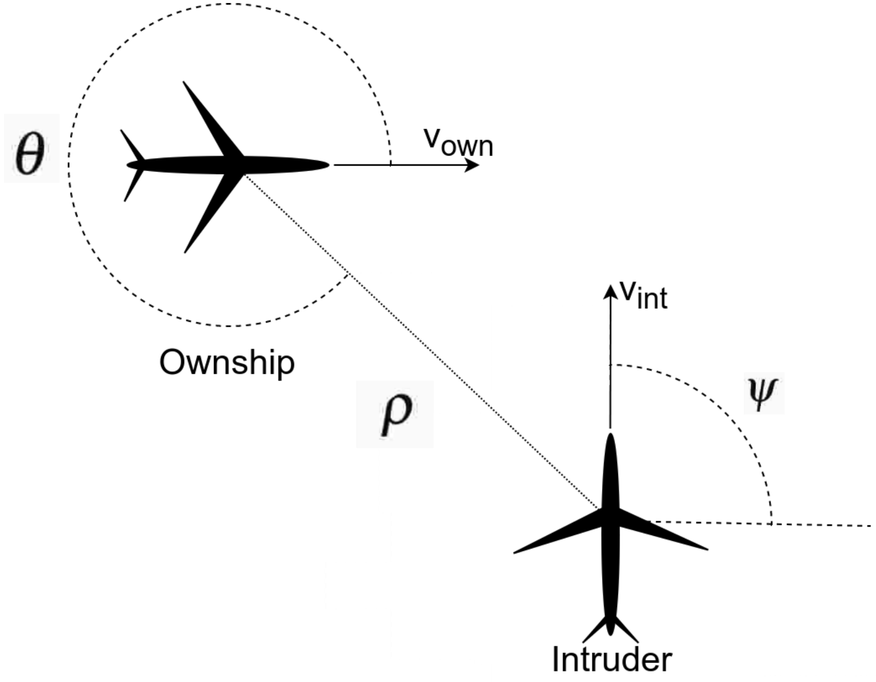
- Planned to be installed on Navy MQ-4C Triton
- Tested by NASA and FAA

Networks

- 5 inputs: $\rho, \theta, \psi, v_o, v_i$
- 5 outputs: COC, weak left, weak right, strong left, strong right
- 6 hidden layers, each with 50 ReLU nodes

Markov decision process

- 45 models=5*9
- previous decision: 5
- time until loss of vertical separation: 9



Sample Example of ACAS Xu Safety Properties

Safety property:

Given input ranges:

$\rho = [1000, 1200]$

$\theta = [\pi, \pi]$

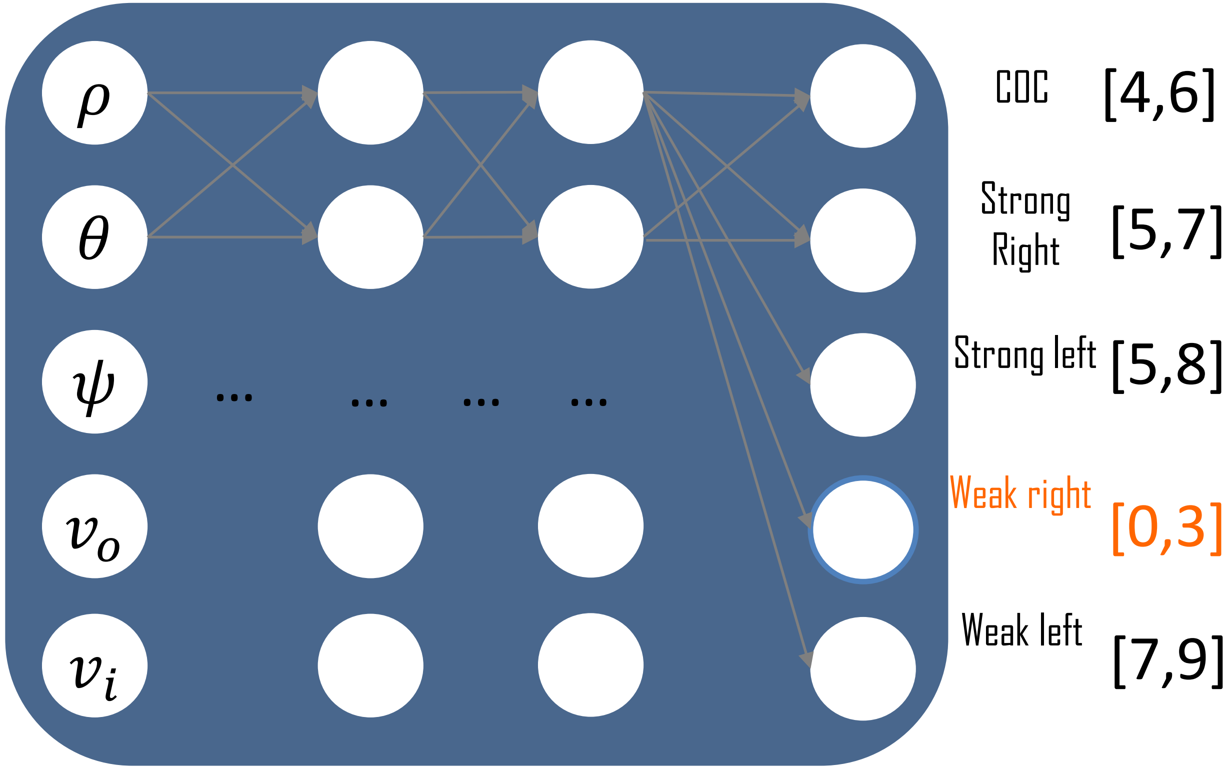
$\psi = [-\pi, -0.5\pi]$

$v_o = [60, 80]$

$v_i = [120, 140]$

Desired output:

Weak right



Verified safe!

ReluVal vs. Reluplex on ACAS Xu

Source	Networks	Reluplex Time (sec)	ReluVal Time (sec)	Speedup
10 Security Properties Proposed by Guy Katz et al.	45	>443,560.73	14,603.27	>30x
	34	123,420.40	117,243.26	1x
	42	35,040.28	19,018.90	2x
	42	13,919.51	441.97	32x
	1	23,212.52	216.88	107x
	1	220,330.82	46.59	4729x
	1	>86400.0	9,240.29	>9x
	1	43,200.01	40.41	1069x
	1	116,441.97	15,639.52	7x
	1	23,683.07	10.94	2165x
5 Additional Security Properties	1	4,394.91	27.89	158x
	1	2,556.28	0.104	24580x
	1	>172,800.0	148.21	>1166x
	2	>172,800.0	288.98	>598x
	2	31,328.26	876.86	36x

Over 200 times faster than Reluplex on average

Neurify [NeuRIPS'18]

Use Linear solver together with symbolic interval analysis

Instead of splitting input features, split inputs to intermediate nodes
- Only relaxed neurons can be overestimated
- Fewer nodes are overestimated
- We can split them instead of inputs

Iteratively split overestimated neurons into two linear cases
- Use linear solver to solve these cases separately

Verification of DAVE Models

DAVE-2 self-driving dataset
- Nvidia end-to-end deep learning for self-driving cars
- Map visual data to steering angles

Convolutional Networks
- 30,000 inputs: raw pixels
- 1 output: steering angle
- Over 10,000 ReLUs

Safety property
- Difference between the predicted steering angle and the ground-truth angle is always less than some threshold

Conclusions and Future Work

Symbolic Interval Analysis is a great tool for verifying NNs
- Efficient and sound over-approximations
- Iterative refinements with bisection
- Can be easily augmented with linear solvers

Exciting new directions
- Use formal verification tools for NNs as part of robust training
- Support verification of complex natural transformations like fog or rain
- Support other activation functions besides ReLUs and other types of networks like RNNs

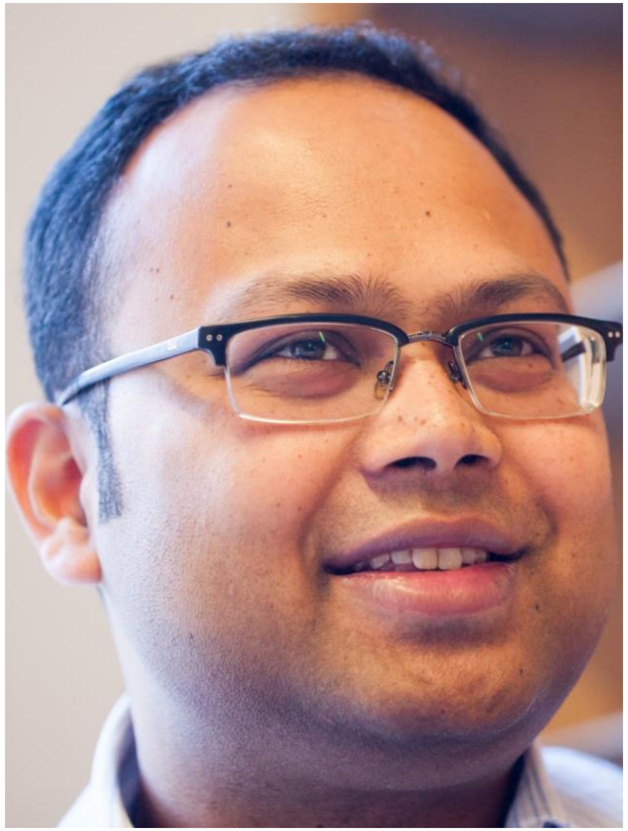
ReluVal: https://github.com/tcwangshiqi-columbia/ReluVal
Neurify: https://github.com/tcwangshiqi-columbia/Neurify

? Questions

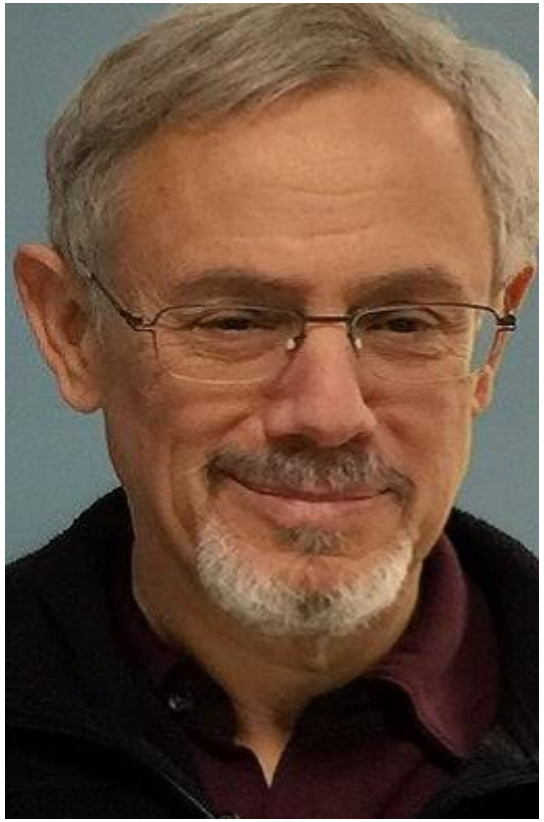
MATHEMATICAL FRONTIERS

Cryptography and Cybersecurity

Kristin Lauter,
Microsoft Research



Suman Jana,
Columbia University



Mark Green,
UCLA (moderator)

MATHEMATICAL FRONTIERS
2019 Monthly Webinar Series, 2-3pm ET

February 12: *Machine Learning for Materials Science*

March 12: *Mathematics of Privacy*

April 9: *Mathematics of Gravitational Waves*

May 14: *Algebraic Geometry*

June 11: *Mathematics of Transportation*

July 9: *Cryptography & Cybersecurity*

August 13: *Machine Learning in Medicine*

September 10: *Logic and Foundations*

October 8: *Mathematics of Quantum Physics*

November 12: *Quantum Encryption*

December 10: *Machine Learning for Text*

Made possible by support for BMSA from the
National Science Foundation
Division of Mathematical Sciences
and the
Department of Energy
Advanced Scientific Computing Research

View webinar videos and learn more about BMSA at www.nas.edu/MathFrontiers