CSI NN

Reverse Engineering of Neural Network Architectures Through Electromagnetic Side Channel

Lejla Batina[1], **Shivam Bhasin[2]**,
Dirmanto Jap[2], Stjepan Picek[3]

[1] Radboud University, Netherlands
[2] NTU, Singapore
[3] TU Delft, Netherlands

USENIX 2019, Santa Clara, USA
13-15 August 2019

Machine Learning & Security

- Machine learning (ML) has wide applications across industries.
- Security is just one popular application for ML
- **US$ 35 Billion Industry by 2024**[1]
- Optimized ML model are Intellectual property
- Leaked models can leak information about sensitive training sets

1 https://www.marketwatch.com/press-release/artificial-intelligence-in-security-market-size-is-projected-to-be-around-us-35-billion-by-2024-2018-10-07

This Work ...

- Reverse Engineering

This Work …

- Reverse Engineering
- Through Side-Channel

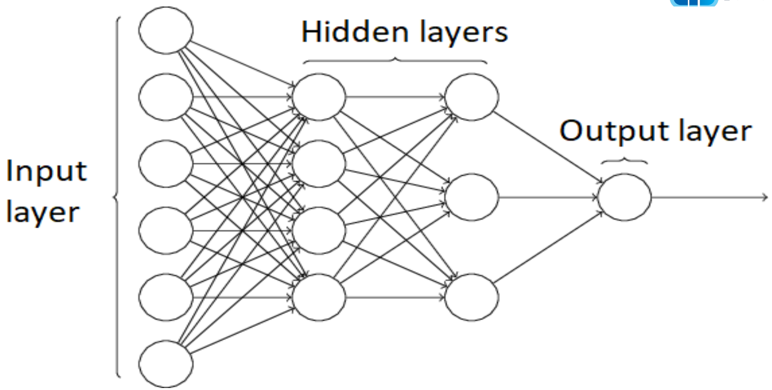
This Work …

- Reverse Engineering
- Through Side-Channel
- Measured by Electromagnetic (EM) probes

This Work …

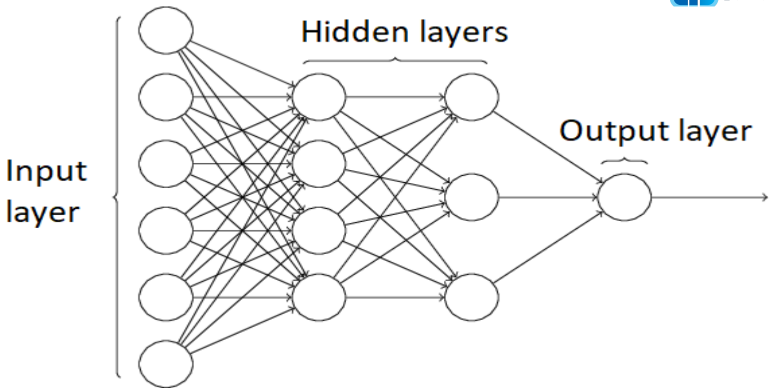
- Reverse Engineering
- Through Side-Channel
- Measured by Electromagnetic (EM) probes
- Of Deep Neural Network (DNN) on embedded devices

This Work …



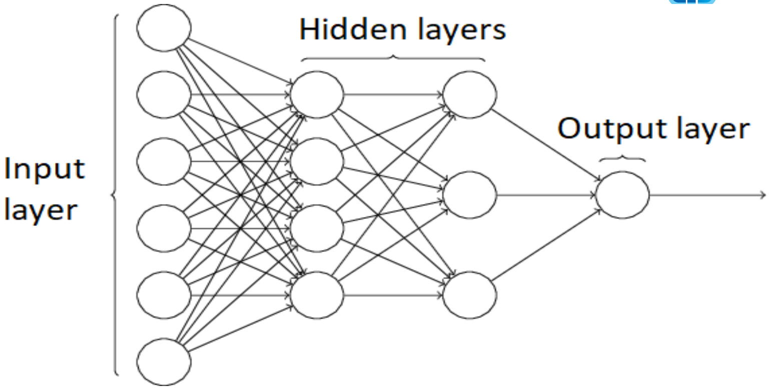
- Reverse Engineering
- Through Side-Channel
- Measured by Electromagnetic (EM) probes
- Of Deep Neural Network (DNN) on embedded devices

This Work …



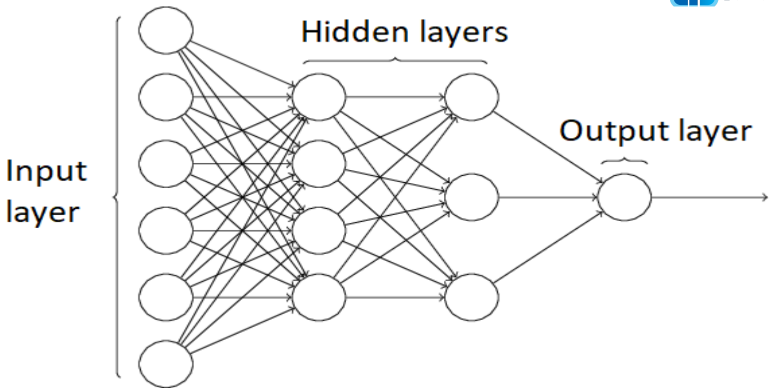
- Reverse Engineering
- Through Side-Channel
- Measured by Electromagnetic (EM) probes
- Of Deep Neural Network (DNN) on embedded devices
- To Recover:

This Work …



- Reverse Engineering
- Through Side-Channel
- Measured by Electromagnetic (EM) probes
- Of Deep Neural Network (DNN) on embedded devices
- To Recover:
  - Number of layers

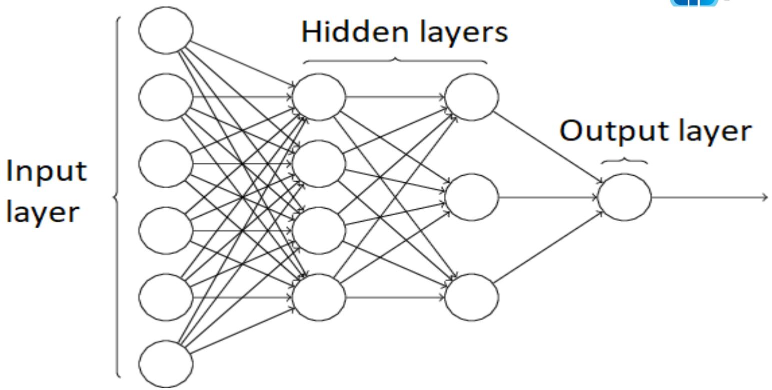
This Work …

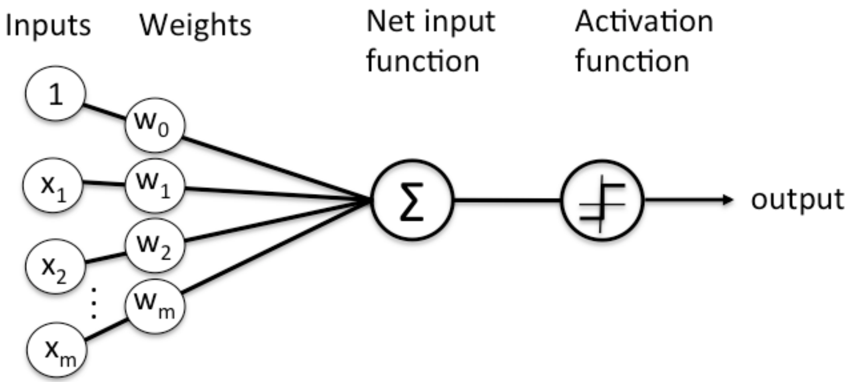


- Reverse Engineering
- Through Side-Channel
- Measured by Electromagnetic (EM) probes
- Of Deep Neural Network (DNN) on embedded devices
- To Recover:
  - Number of layers
  - Number of neurons in each layer

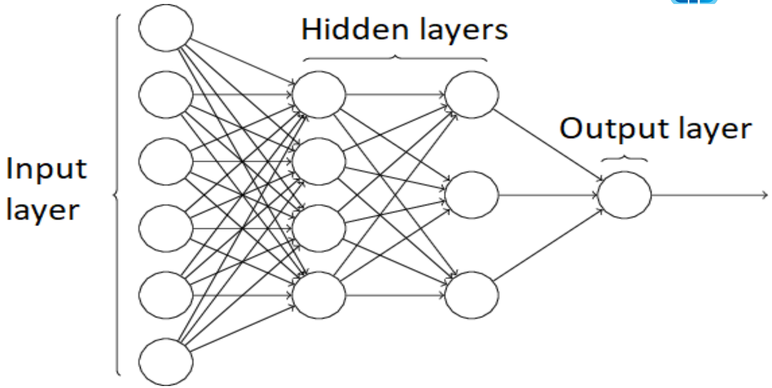
This Work …

- Reverse Engineering
- Through Side-Channel
- Measured by Electromagnetic (EM) probes
- Of Deep Neural Network (DNN) on embedded devices
- To Recover:
  - Number of layers
  - Number of neurons in each layer

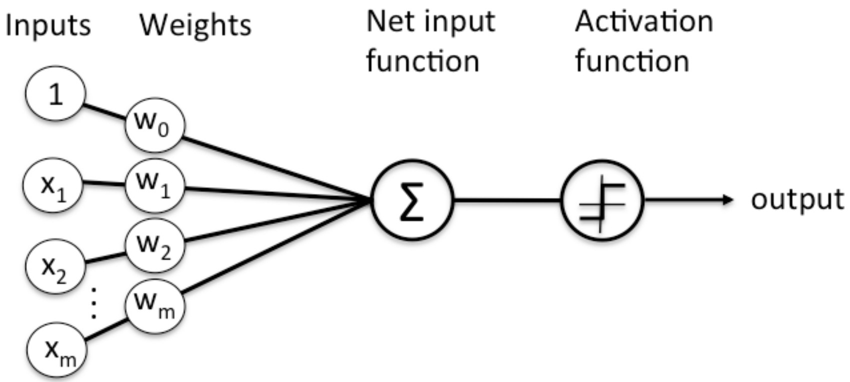




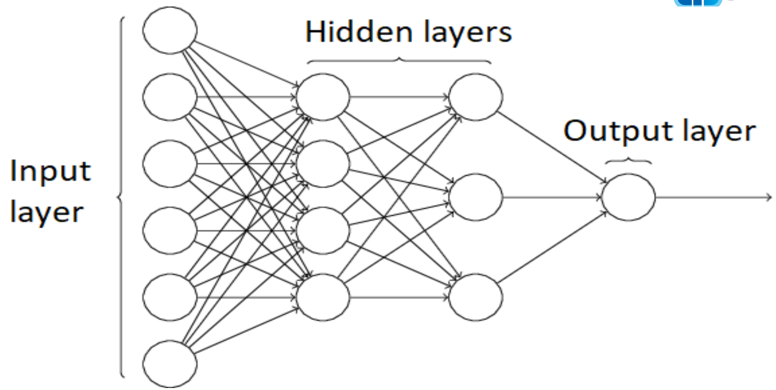
This Work …



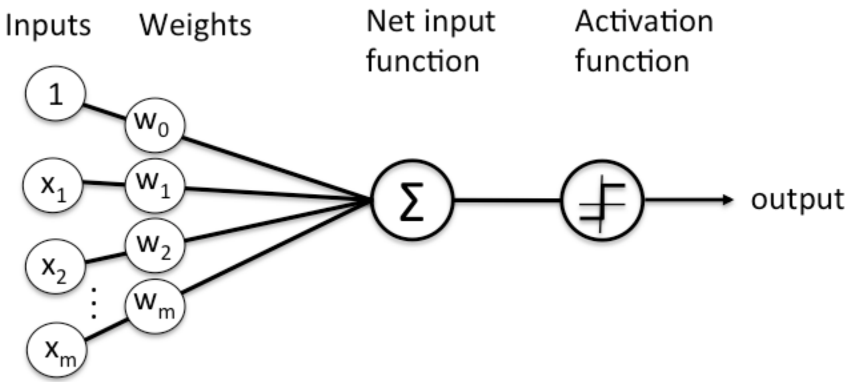
- Reverse Engineering
- Through Side-Channel
- Measured by Electromagnetic (EM) probes
- Of Deep Neural Network (DNN) on embedded devices
- To Recover:
  - Number of layers
  - Number of neurons in each layer
  - Activation function in each neuron



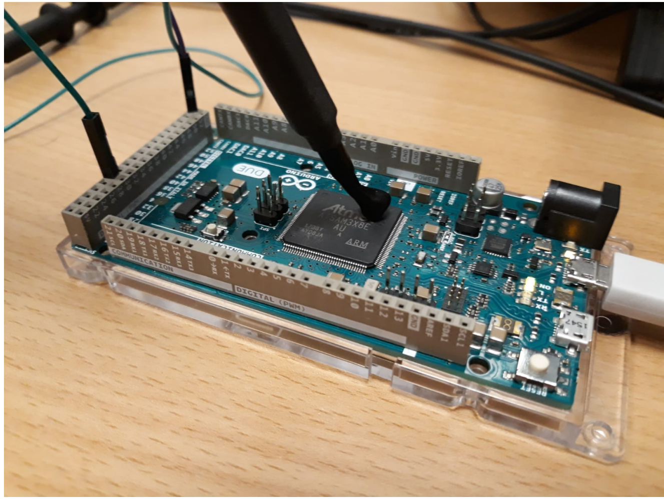
This Work …



- Reverse Engineering
- Through Side-Channel
- Measured by Electromagnetic (EM) probes
- Of Deep Neural Network (DNN) on embedded devices
- To Recover:
  - Number of layers
  - Number of neurons in each layer
  - Activation function in each neuron
  - Input weights to each neuron

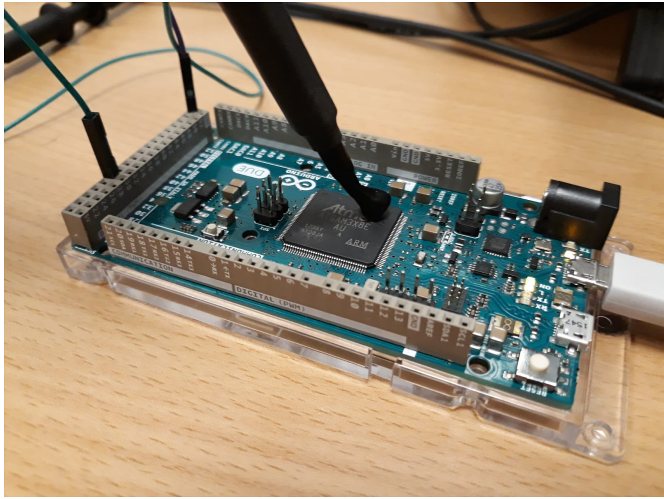


Electromagnetic (EM) SCA



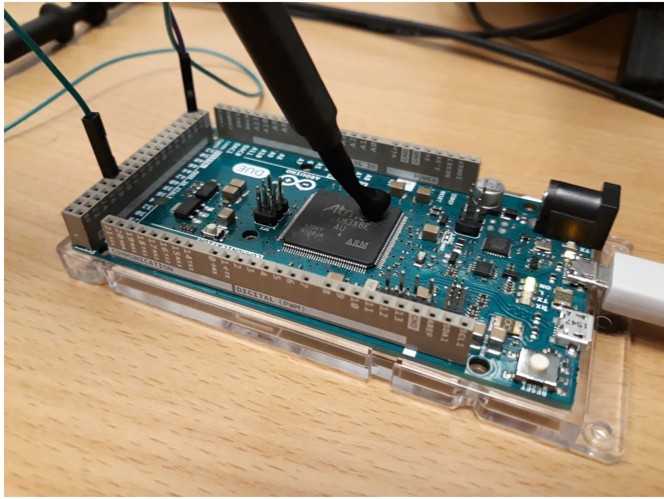
Electromagnetic (EM) SCA

- Non-invasive



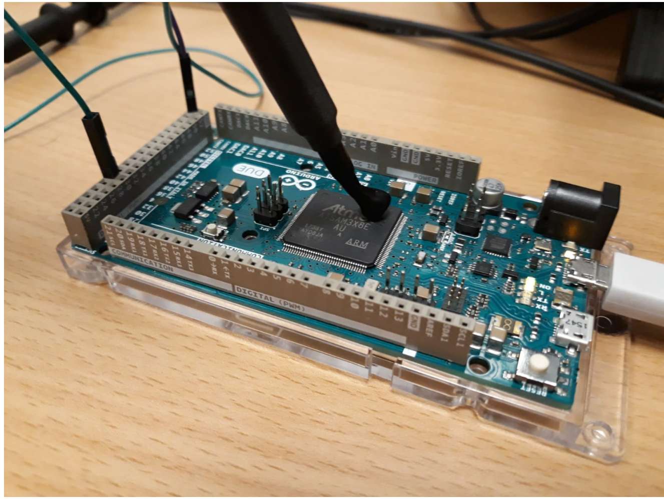
Electromagnetic (EM) SCA

- Non-invasive
- Serious threat to pervasive computing



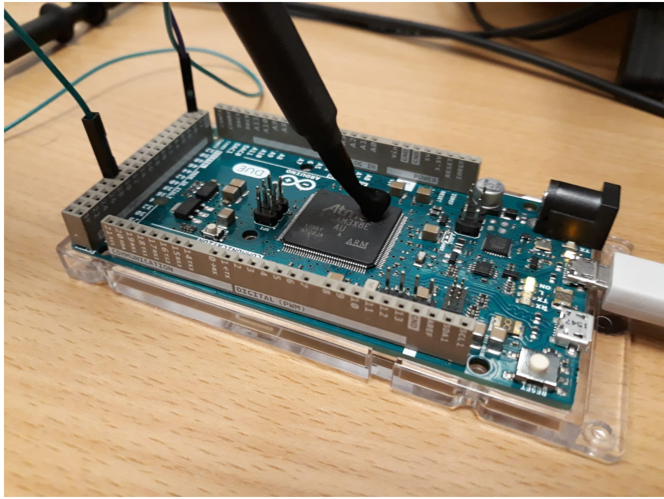
Electromagnetic (EM) SCA

- Non-invasive
- Serious threat to pervasive computing
- Exploiting unintentional EM leakage

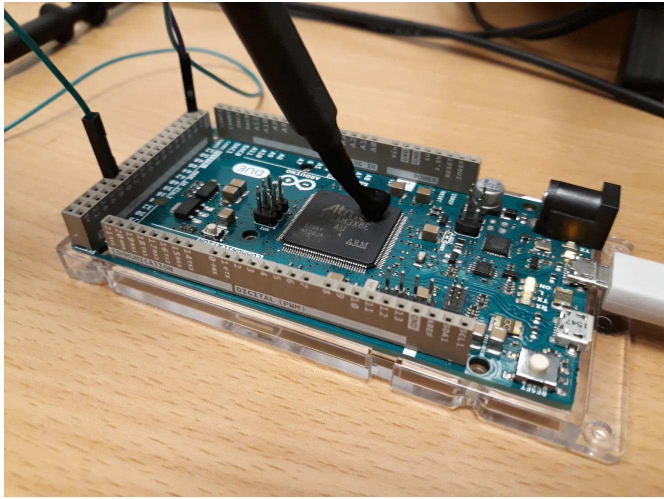


Electromagnetic (EM) SCA

- Non-invasive
- Serious threat to pervasive computing
- Exploiting unintentional EM leakage
- Powerful & practical
  - Keeloq
  - FPGA Bitstream encryption
  - Bitcoin wallets



Electromagnetic (EM) SCA

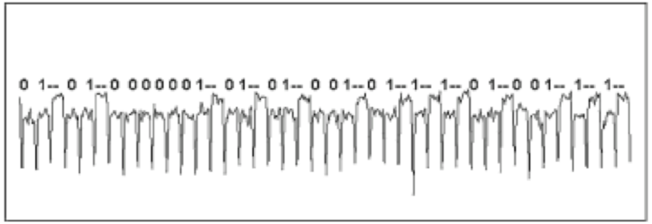


- Non-invasive
- Serious threat to pervasive computing
- Exploiting unintentional EM leakage
- Powerful & practical
  - Keeloq
  - FPGA Bitstream encryption
  - Bitcoin wallets
- Applications beyond secret key recovery

Electromagnetic (EM) SCA

Simple EM Analysis (SEMA)

- Adversary learns secret information by visual inspection of (usually single) power/EM measurement
- Ex: observe square & multiply in exponentiation etc.



Differential EM Analysis (DEMA)

- Adversary extract secret information **statistically** from EM trace
- Target leakage from function f(x,k) of Secret **k,** input **x**
- EM leakage ➔ **L(f(x,k))**
- Correct key **k*** maximizes: ρ**(t, L(f(x,k)))**
- Most commonly used leakage model L is **Hamming Weight (HW)**
- A microcontroller leaks in **Hamming Weight** when sensitive data is loaded to pre-charged data bus

Source: https://anysilicon.com/side-channel-attacks-differential-power-analysis-dpa-simple-power-analysis-spa-works/

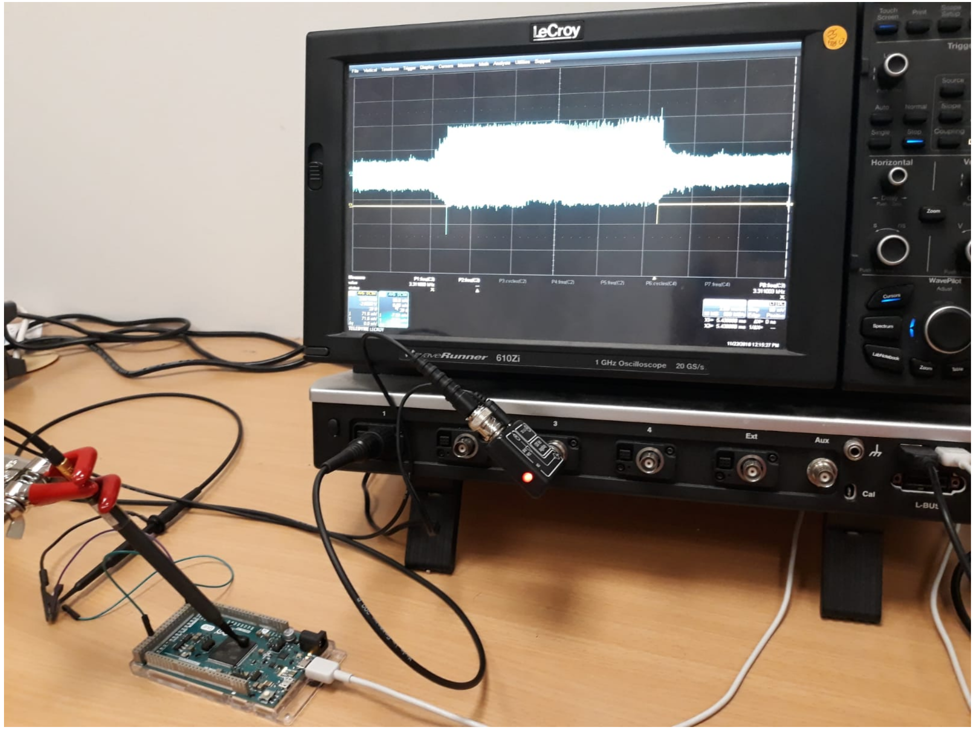
Adversary Model

- Recover the neural network architecture using only side-channel information
- Adversary does not know the architecture of the used network but can feed random/known inputs to the DNN and capture corresponding electromagnetic side-channel traces
- No assumption on the type of inputs; we work with real numbers
- **Assumption**: Implementation of the machine learning algorithm with no side-channel countermeasures

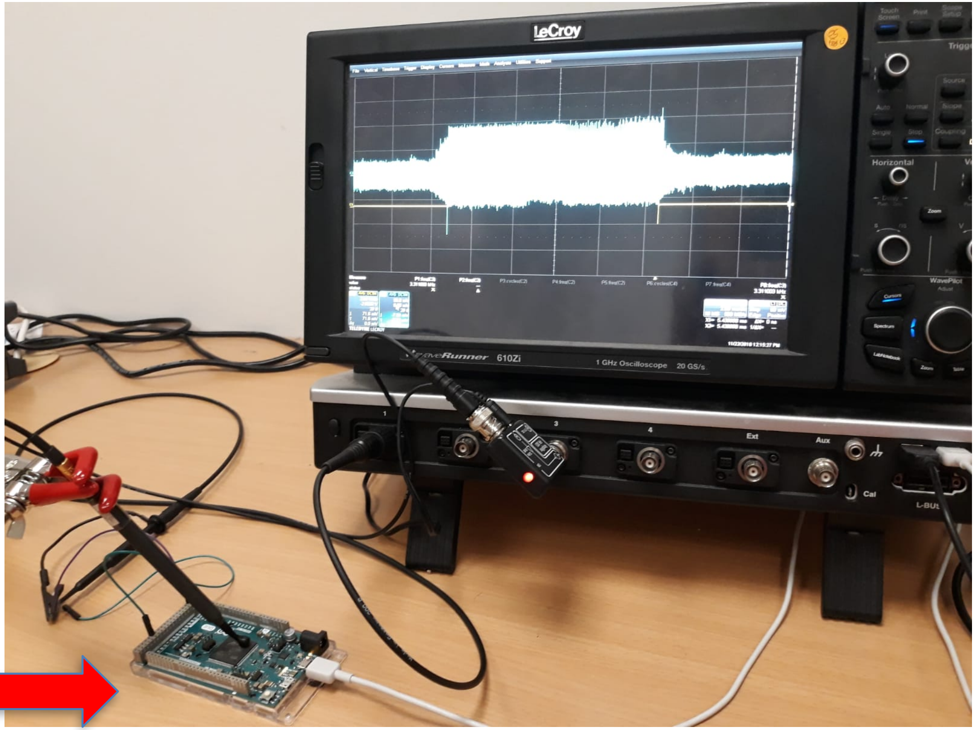
Experimental Setup

- Passive EM Measurement
- Near-field probe
- 30dB pre-amplifier for clear signal
- Measurements averaged for noise filtering
- For bigger networks, measurements are made sequentially for different layers
- Targets: ATMEGA AVR328P, ARM Cortex-M3

Experimental Setup

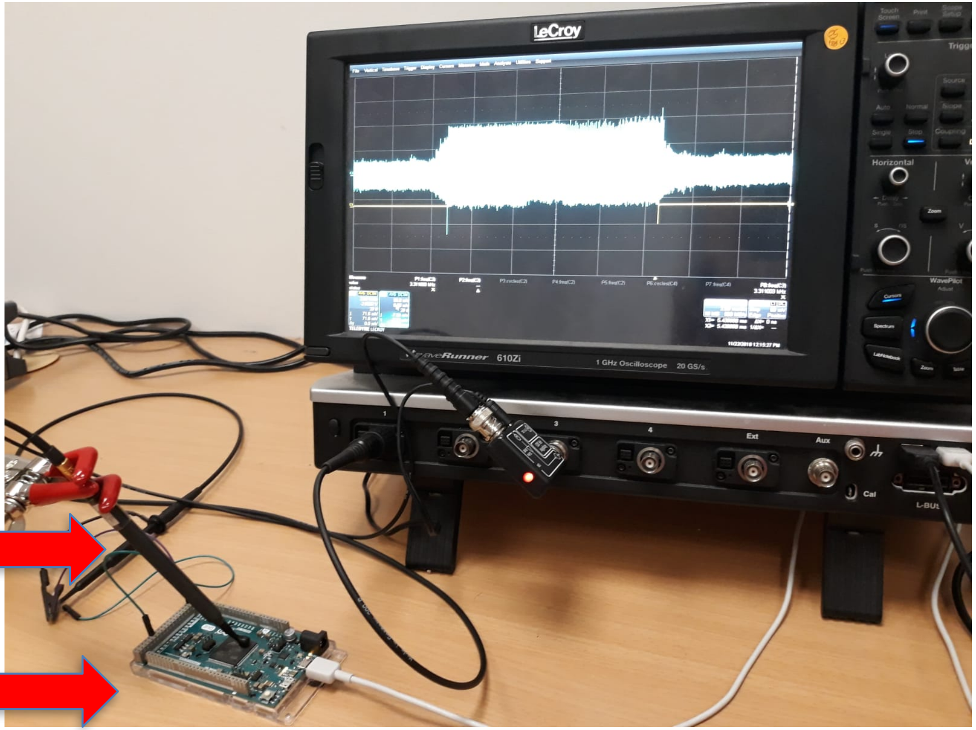


Experimental Setup



Target

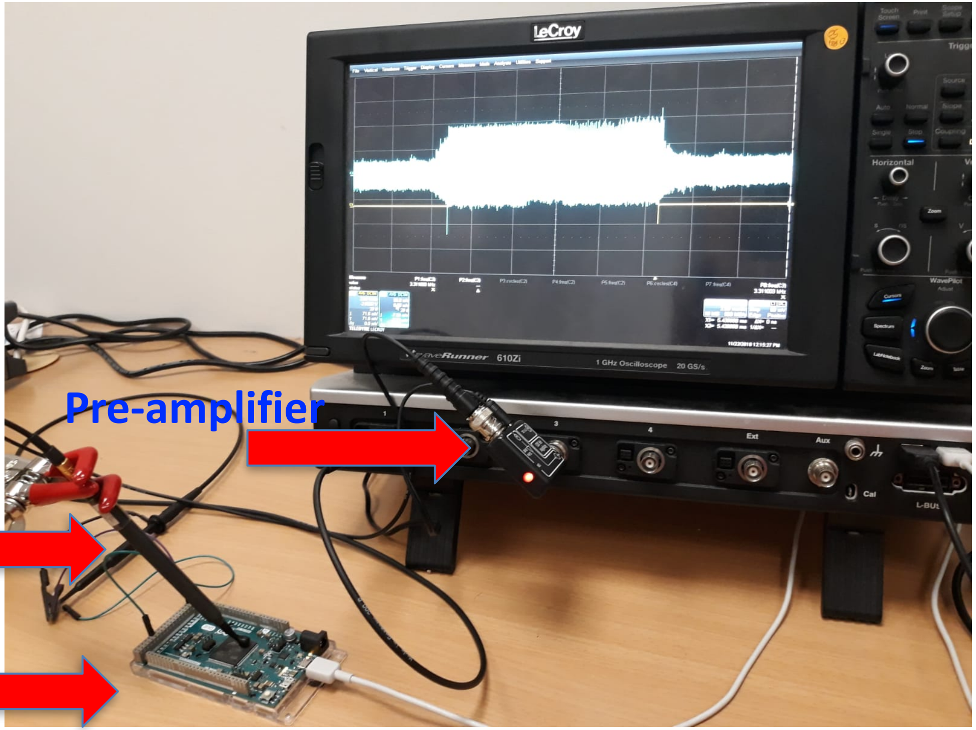
Experimental Setup



Probe

Target

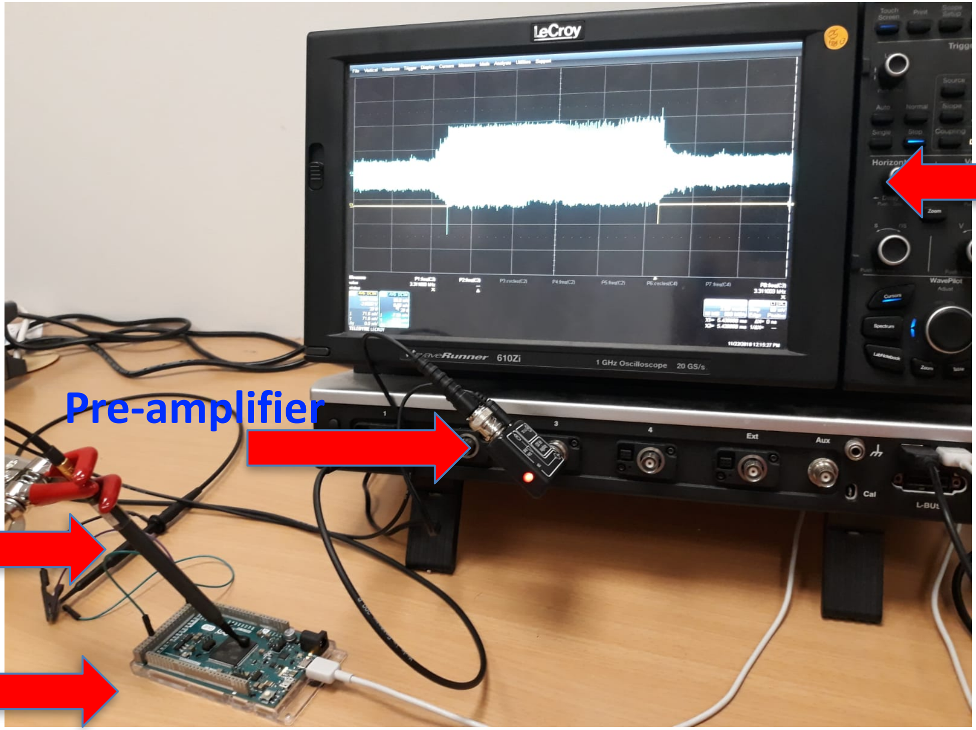
Experimental Setup



Probe

Target

Experimental Setup



Pre-amplifier

Trace

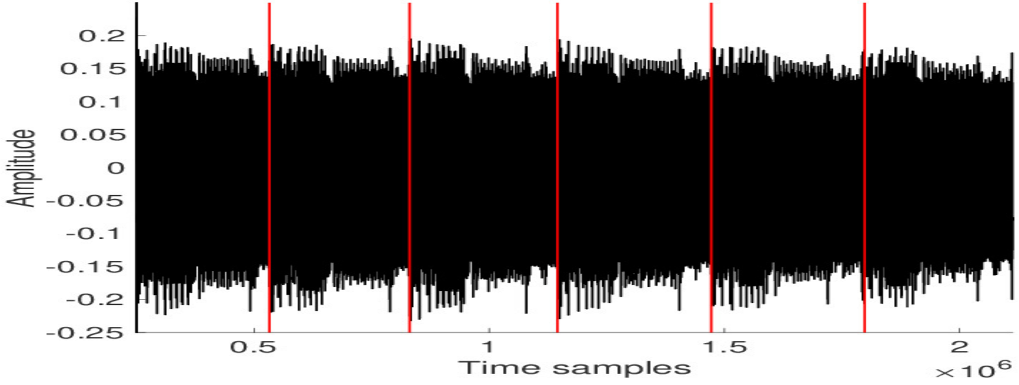
Probe

Target

Lets Start With Some Visual Inspection!!!!

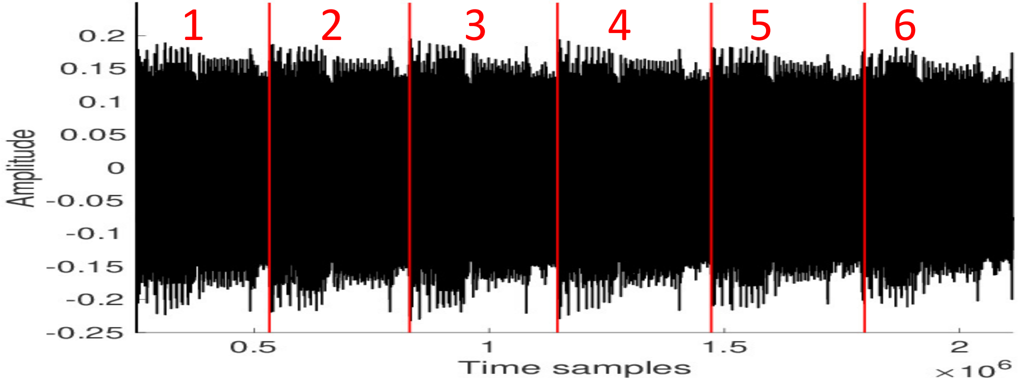
Identifying Neurons

- **Simple EM Analysis**
- Hidden layer with 6 neurons = 6 repeating patterns
- Each neuron executes a series of multiplication, followed by activation
- Activation Function in this case = Sigmoid



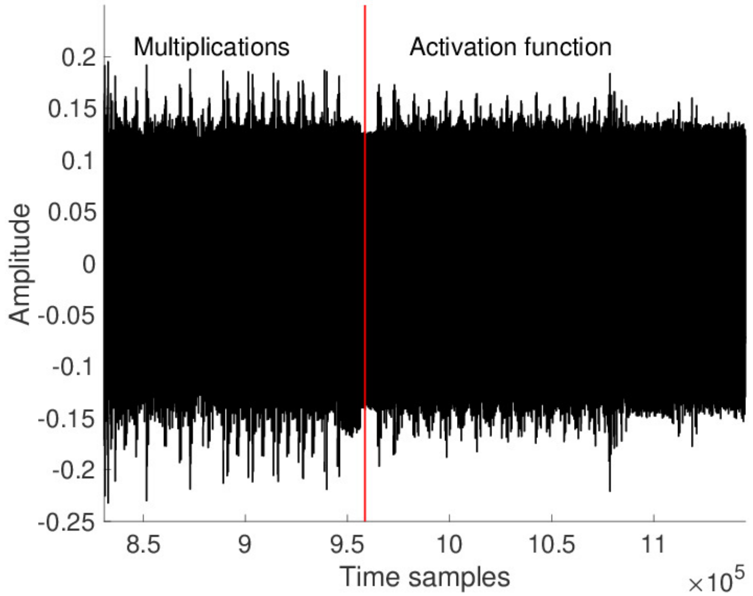
Identifying Neurons

- **Simple EM Analysis**
- Hidden layer with 6 neurons = 6 repeating patterns
- Each neuron executes a series of multiplication, followed by activation
- Activation Function in this case = Sigmoid



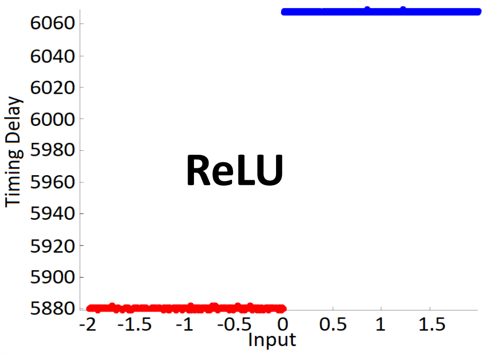
Recovering Activation Function

- **Timing Attack**
- Each activation function has distinct timing pattern
- Timing patterns can be pre-characterized for different NN libraries
- We measure **precise timing** of activation function using **EM measurement** on oscilloscope.

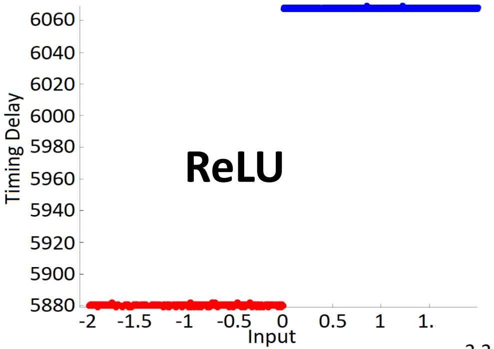


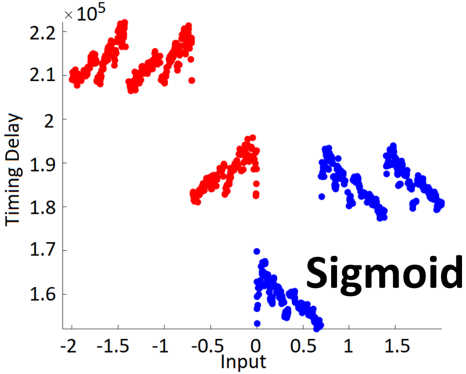
Timing Patterns of Various Activation Function

Timing Patterns of Various Activation Function

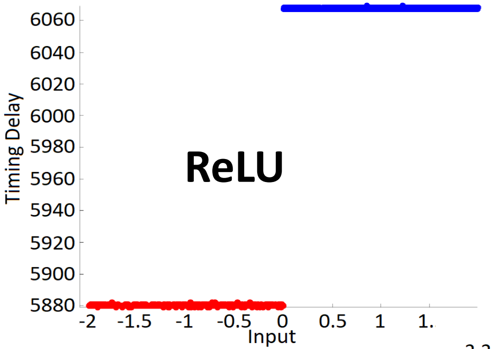


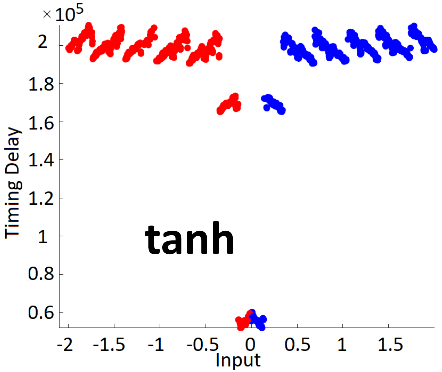
Timing Patterns of Various Activation Function

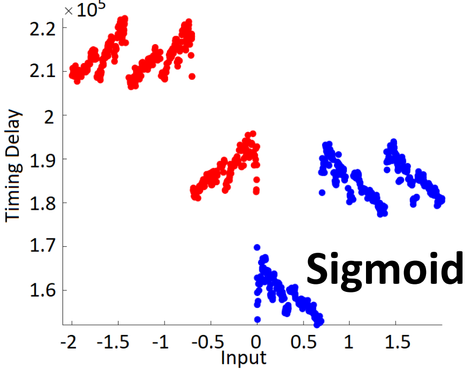




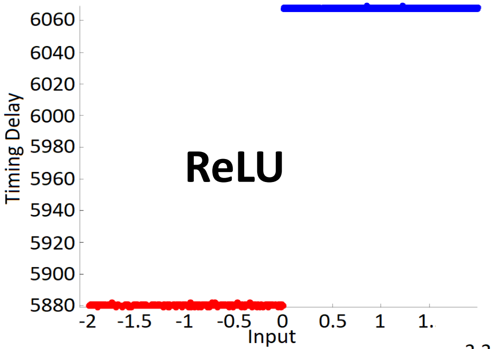
Timing Patterns of Various Activation Function

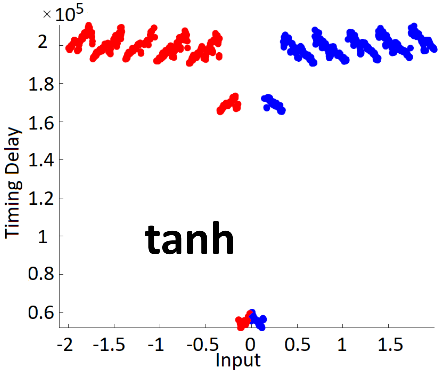


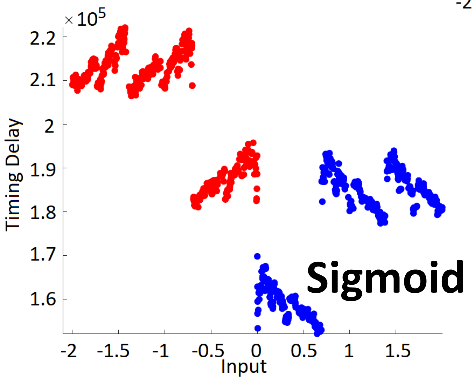


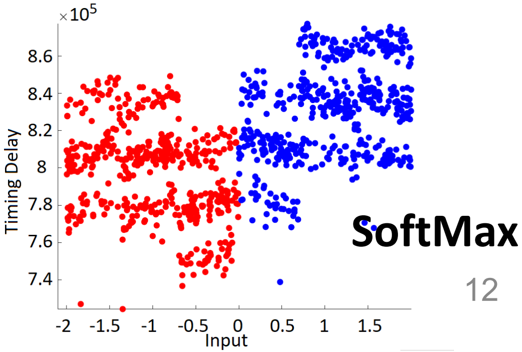


Timing Patterns of Various Activation Function

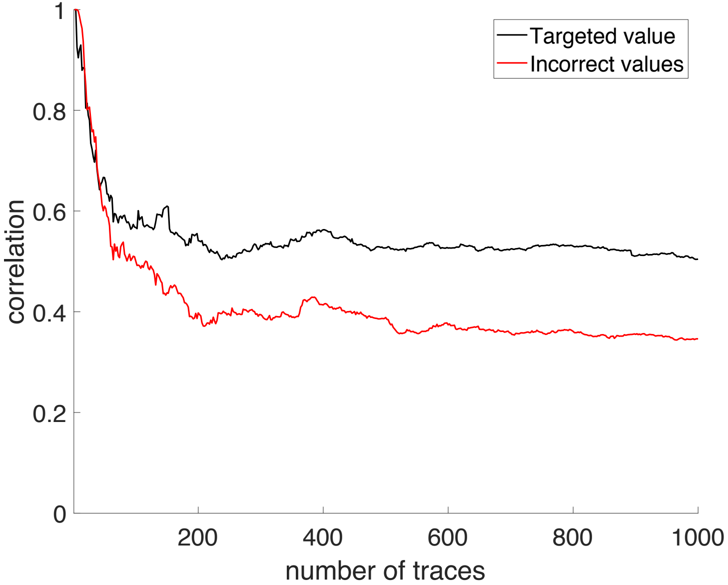




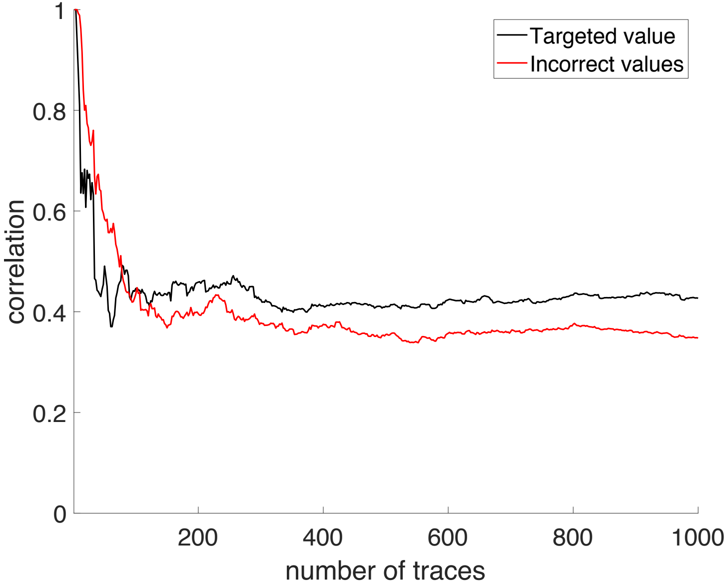




Recovering Weights

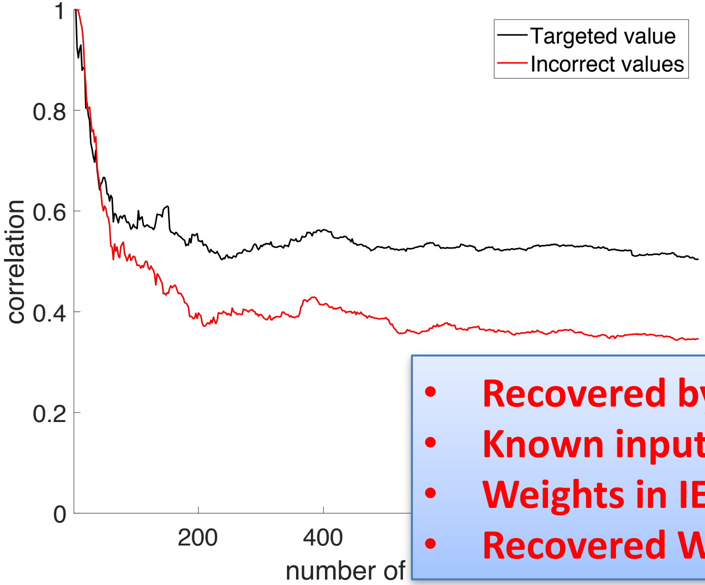


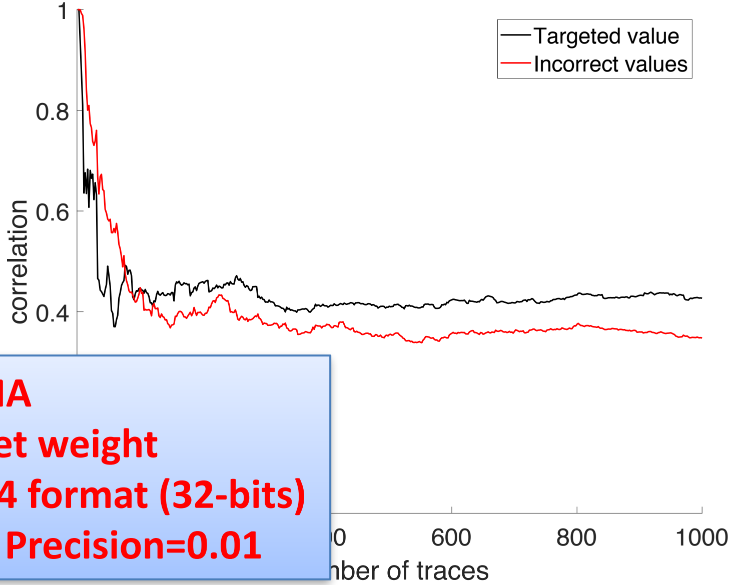
First byte recovery
(sign and 7-bit exponent)



Second byte recovery
(lsb exponent and mantissa)

Recovering Weights



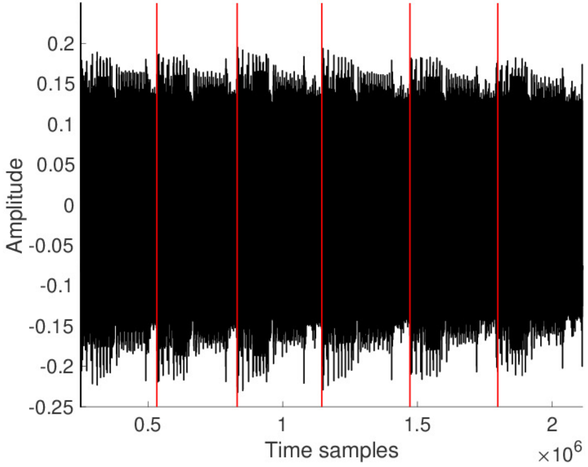


- **Recovered by DEMA**
- **Known input, secret weight**
- **Weights in IEEE 754 format (32-bits)**
- **Recovered Weight Precision=0.01**

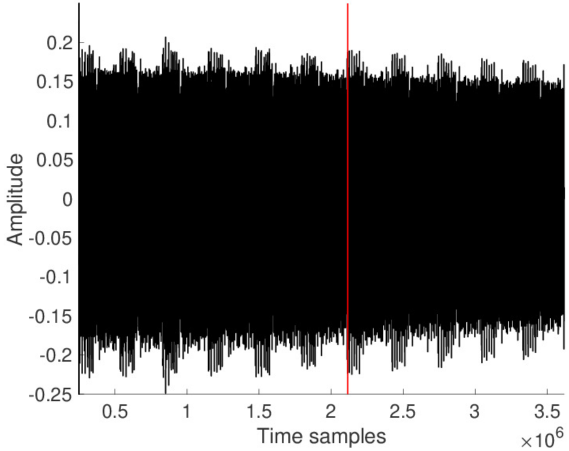
First byte recovery
(sign and 7-bit exponent)

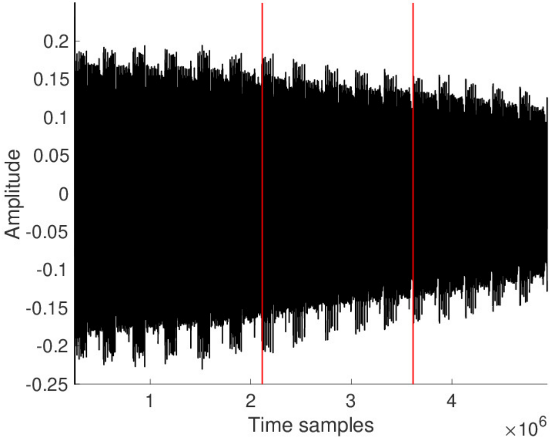
Second byte recovery
(lsb exponent and mantissa)

Recovering Number of Neurons & Layers

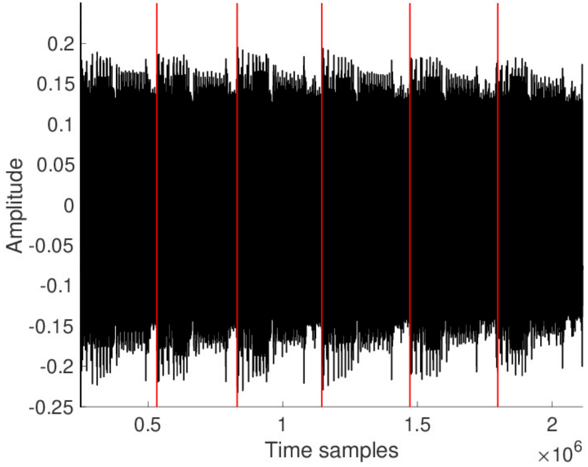


One hidden layer
6 neurons



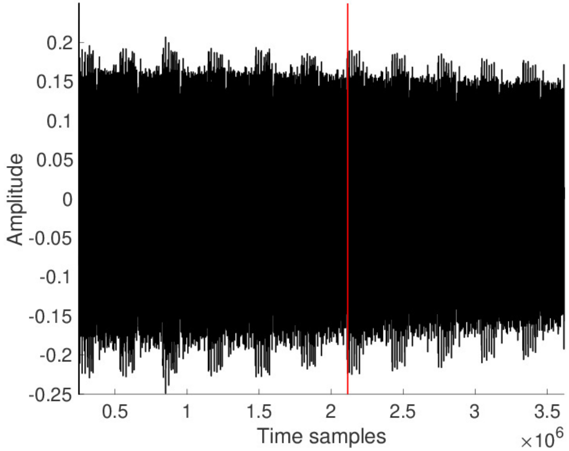


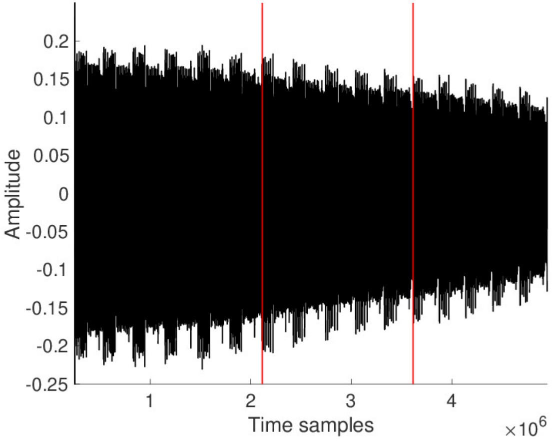
Recovering Number of Neurons & Layers



One hidden layer
6 neurons

Two hidden layer
(6,5) neurons

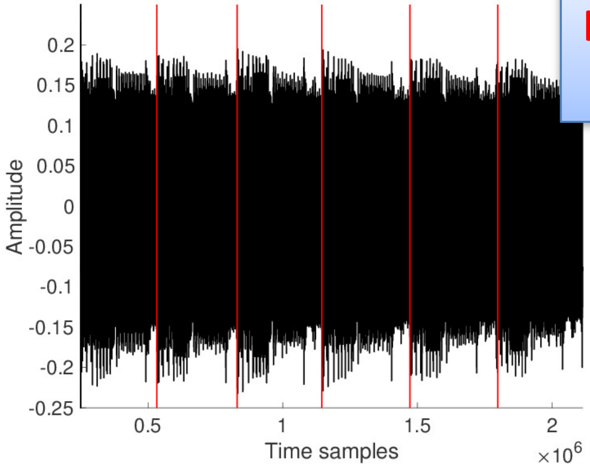




Three hidden layer
(6,5,5) neurons

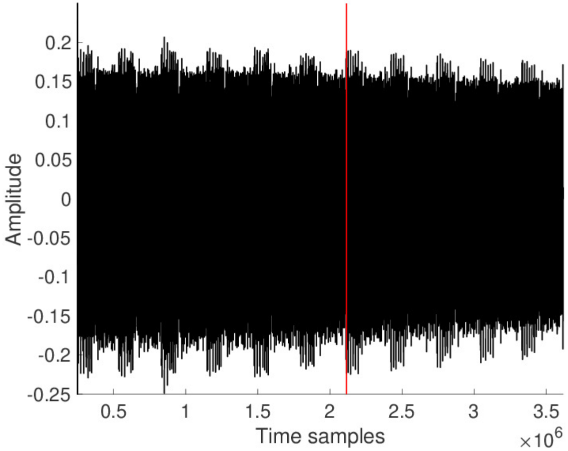
Recovering Number of Neurons & Layers

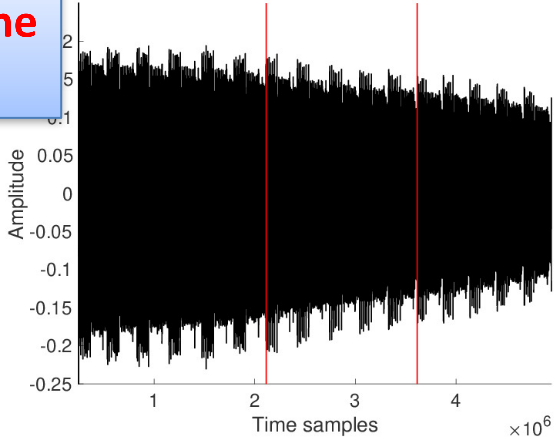
DEMA on weights used to determine layer boundaries



One hidden layer
6 neurons

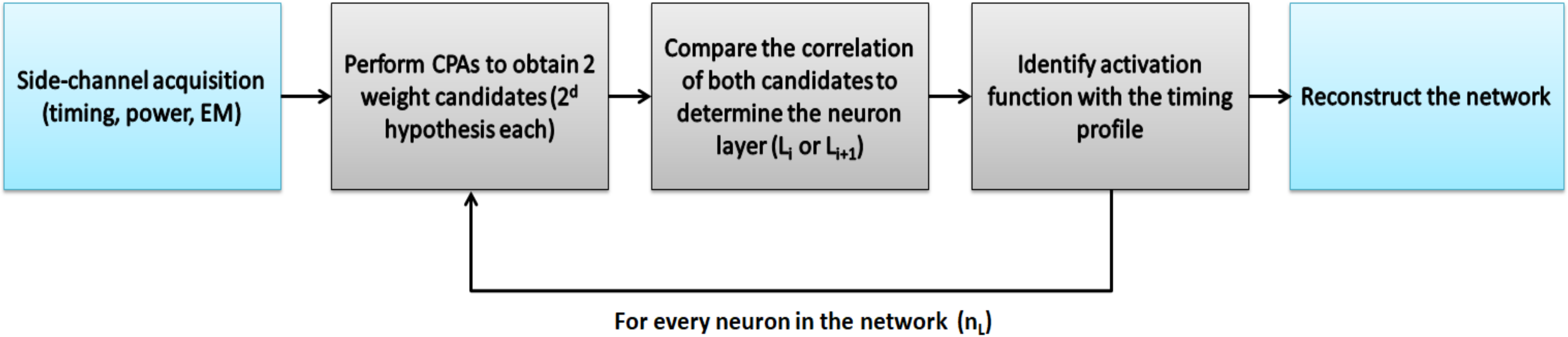
Two hidden layer
(6,5) neurons



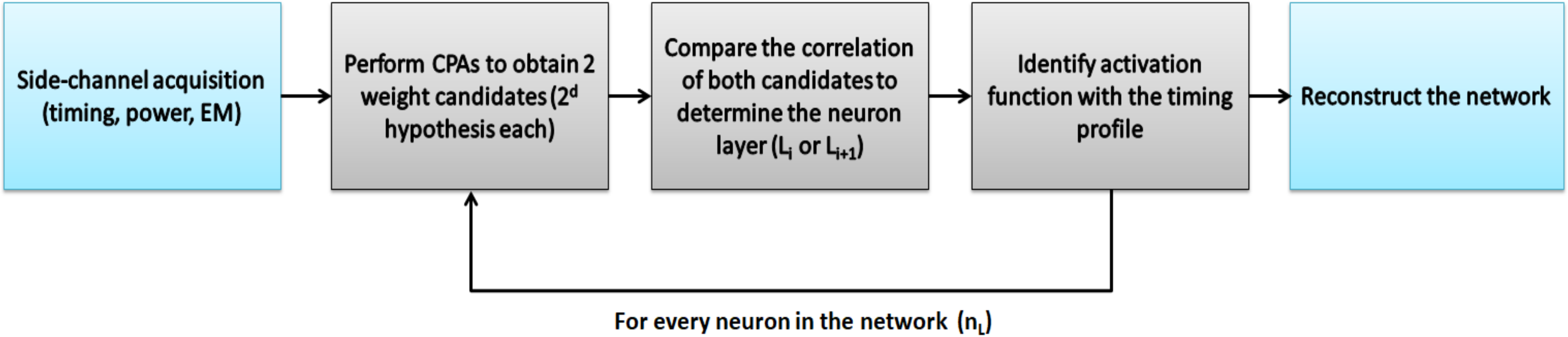


Three hidden layer
(6,5,5) neurons

Full Network Recovery

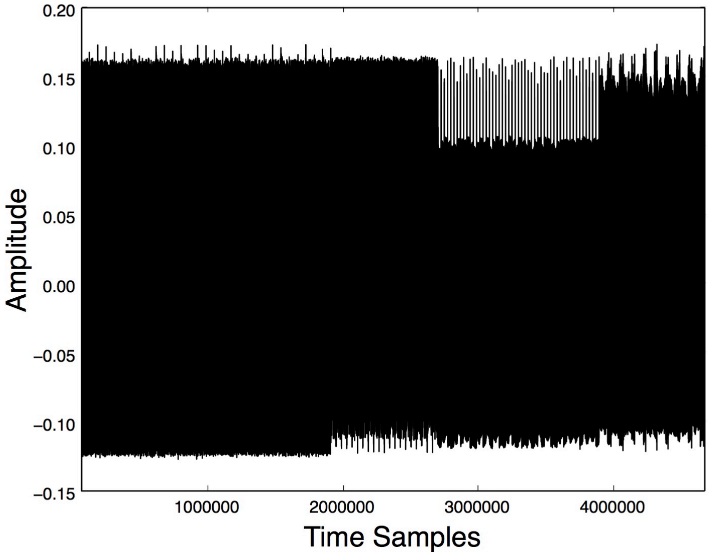


Full Network Recovery

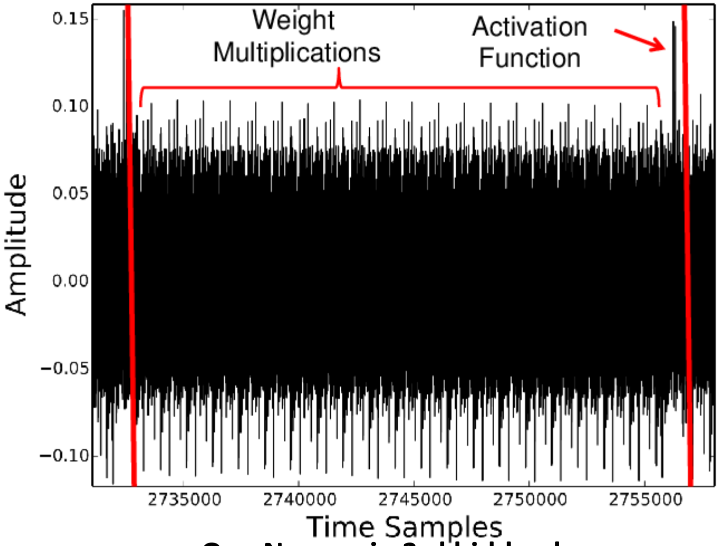


Recovery is performed layer by layer, neuron by neuron.
One neuron at a time, starting from input layer

Results on ARM Cortex-M3

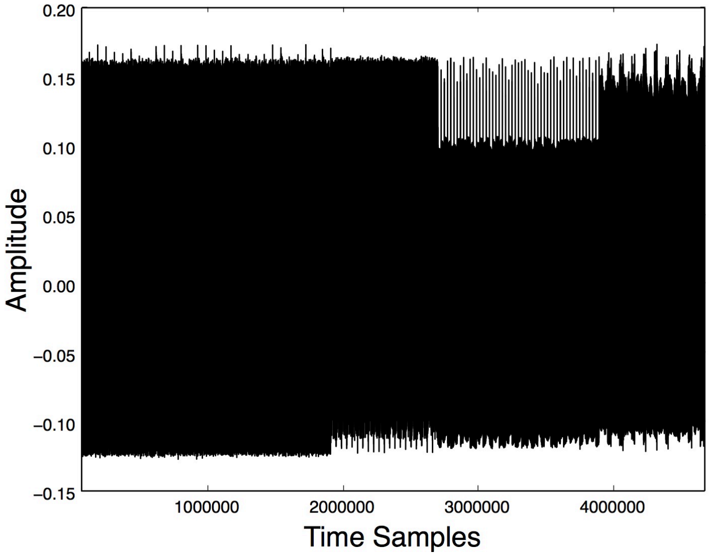


Four hidden layer
(50,30,20,50) neurons

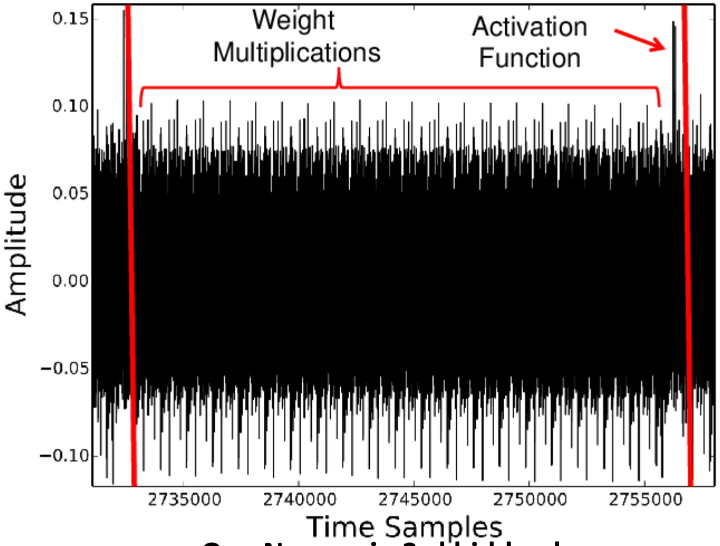


One Neuron in 3rd hidden layer
20 multiplications, 1 ReLU

Results on ARM Cortex-M3



Four hidden layer
(50,30,20,50) neurons

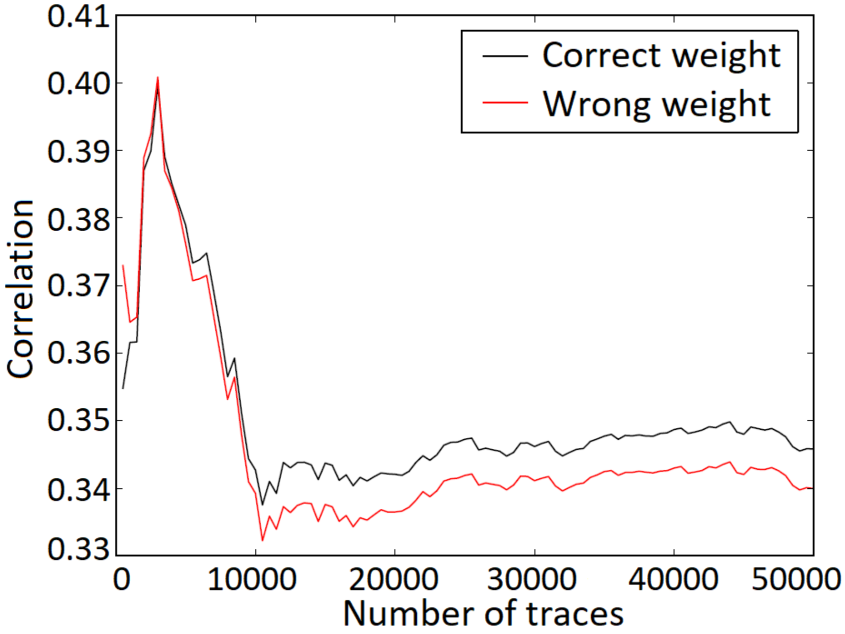


One Neuron in 3rd hidden layer
20 multiplications, 1 ReLU

With MNIST: Accuracy 98.16% (original) vs 98.15% (reverse engineered)
Average weight error: 0.0025.

Extension to CNN on ARM Cortex-M3

- CIFAR-10 dataset.
- Target the multiplication operation from the input with the weight, similar as in previous experiments.
- fixed-point arithmetic (8-bits).
- The original accuracy of the CNN equals 78.47% and the accuracy of the recovered CNN is 78.11%.



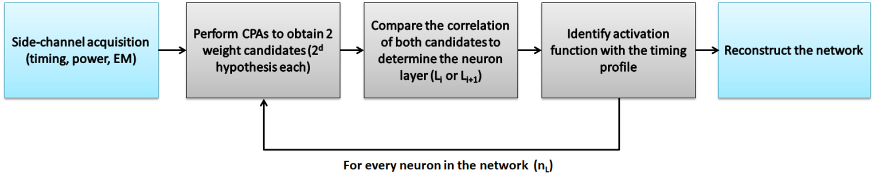
Conclusions

- With an appropriate combination of SEMA and DEMA techniques, all sensitive parameters of the network can be recovered.
- A serious threat to commercial NN IPs
- The attack methodology scales linearly with the size of the network.
- The proposed attacks are both generic in nature and more powerful than the previous works in this direction.
- Can be adapted for recovery of sensitive training/testing data
- SCA countermeasures (masking/hiding) would help but overhead will be too high for NN. Motivates research for optimised countermeasures.

Thank You !!!

Questions ???

Full Network Recovery



- The combination of previously developed individual techniques can thereafter result in full reverse engineering of the network.
- Recovery is performed layer by layer, neuron by neuron, one at a time.
- Complexity grows linearly with network size.
- The first step is to recover the weight w_{L0} of each connection from the input layer (L_0) and the first hidden layer (L_1).
- In order to determine the output of the sum of the multiplications, the number of neurons in the layer must be known.
- Using the same set of traces, timing patterns for different inputs to the activation function can be built.
- The same steps are repeated in the subsequent layers

Recovering Weights

- Correlation Power Analysis (CPA) i.e., a variant of DPA using the Pearson's correlation as a statistical test.
- CPA targets the multiplication $m = x \cdot w$ of a known input x with a secret weight w.
- Using the HW model, the adversary correlates the activity of the predicted output m for all hypothesis of the weight, with side-channel trace t
- The correct value of the weight w will result in a higher correlation standing out from all other wrong hypotheses w_*, given enough measurements.
- As data is represented in IEEE 754 format, each floating point number is 32 bits. 1 sign bit, 8 exponent bits and 23 mantissa bits
- Exact weight recovery is not required but only up to a precision (we choose 0.01)