Compressed Deep Neural Networks and their Fault Resiliency

MAJID SABBAGH

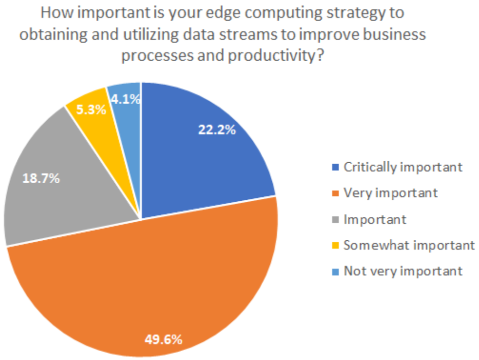
AAISS - SUMMER 1398/2019

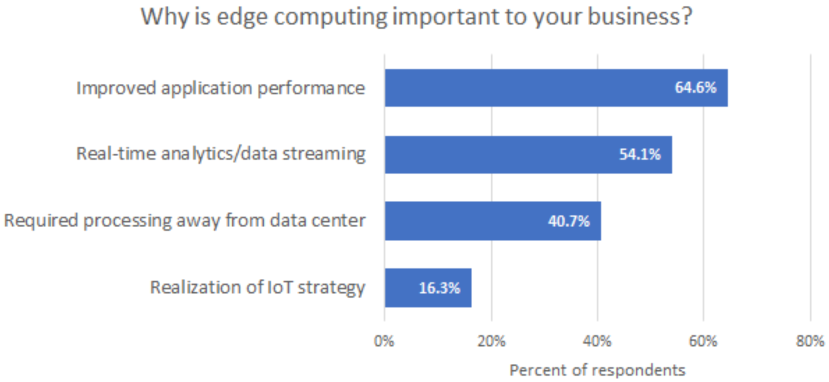
Computation enterprise dynamics

- 80% of enterprises will have shut down their traditional data centers by 2025 vs. 10% in 2018 (Gartner report)
- Computing workloads migrating:
  - On-premises data centers → the cloud
  - **Cloud data centers → "edge" locations, i.e. closer to the source of data**

Why edge computing is important?

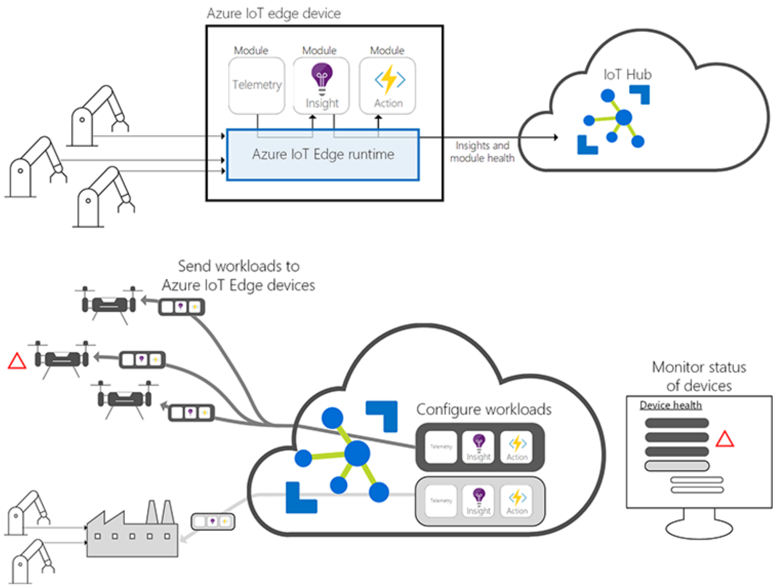
A survey done over 500 North American companies, ranging from 500 to 50,000 employees, by Futurum Research:

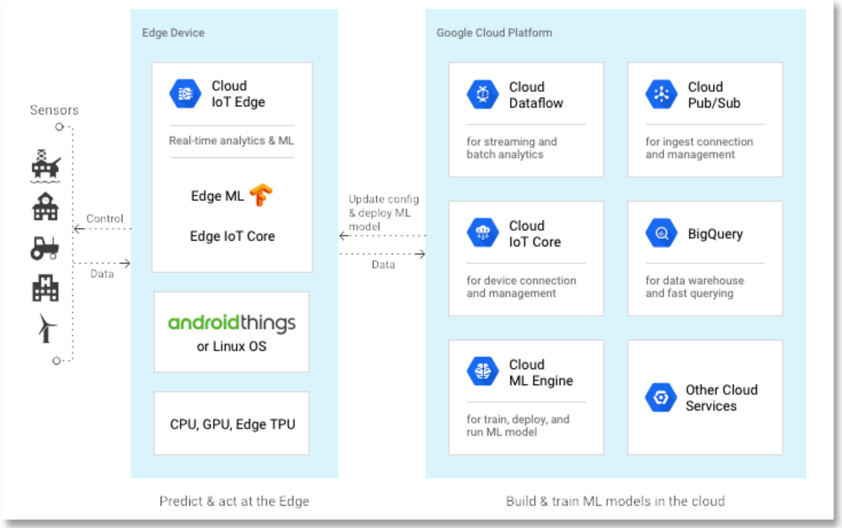




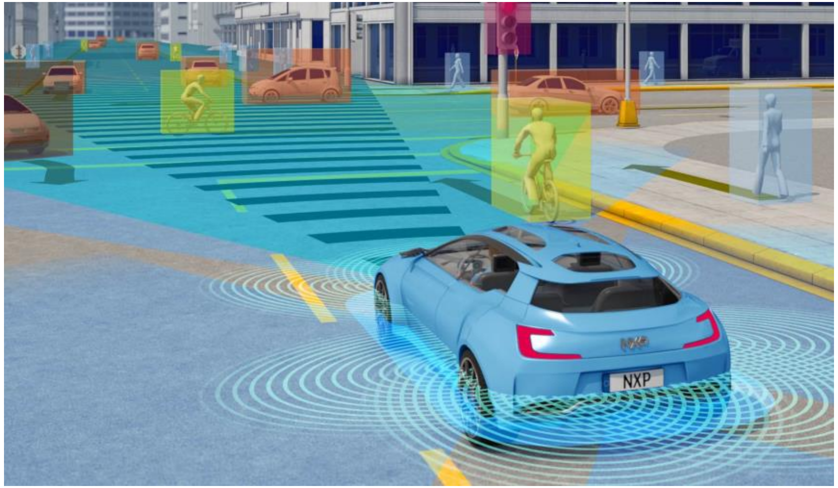
Data: Futurum Research / Charts: ZDNet

Also the big players' direction…





Autonomous cars, a great example!



Advantages of computing at the edge

Latency: ***Reduced***

- Shortening the distance the data has to travel
- Reducing the granularity of upstream data

Bandwidth: ***Less needed***

- Less occupation of the upstream bandwidth (send up only major reports, receive only commands and updates)
- Small sized data are consumed and processed

Security (or rather privacy): *If implemented correctly,* ***enhances privacy, but for other security aspects it's hard to say***

Problem Statement

Deep neural networks (DNNs) are very popular:

- E.g., image classification and video analysis for autonomous cars

Integrity (fault resiliency) of DNNs is critical:

- Reliability – system availability
- Security – data confidentiality and IP protection
- Safety – fault-free operation

Many variants (different architectures, compression, etc.)

Deployed on different platforms (CPU, GPU, FPGA, etc.)

→ *A need to evaluate their impact on the fault resiliency*

Research aims

Developing a simulation framework for assessing fault resiliency of DNN models

- Comparing the fault resiliency of different DNN layers and different data types
- Evaluating the effect of DNNs model compression on the fault resiliency

"Evaluating Fault Resiliency of Compressed Deep Neural Networks", ICESS'19

Majid Sabbagh, Cheng Gongye, Yunsi Fei, Yanzhi Wang

Outline

Background

- DNN layers
- DNN model compression and data types
- Storage faults

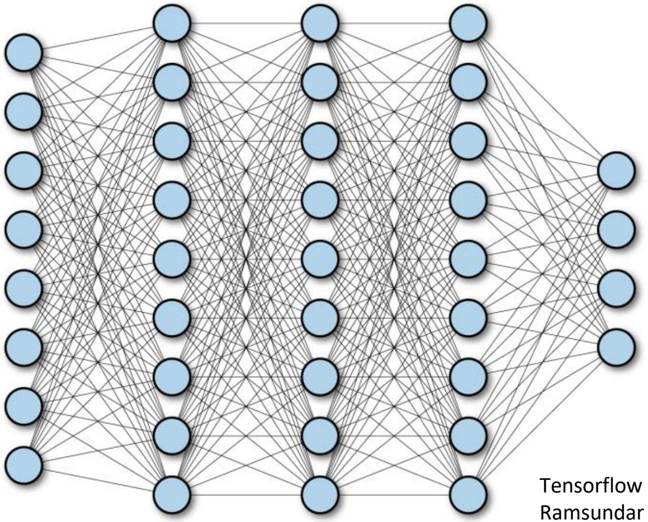
Experimental setup

- Models
- Evaluation Metric
- Procedure

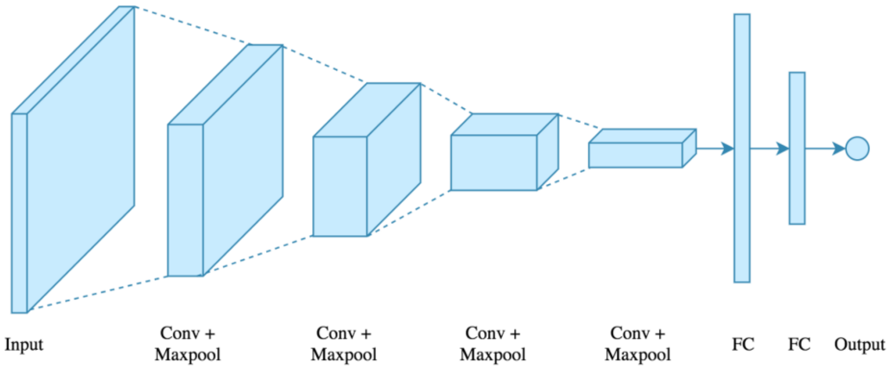
Results

Future work and conclusion

DNN layers



Tensorflow for Deep Learning
Ramsundar et al.

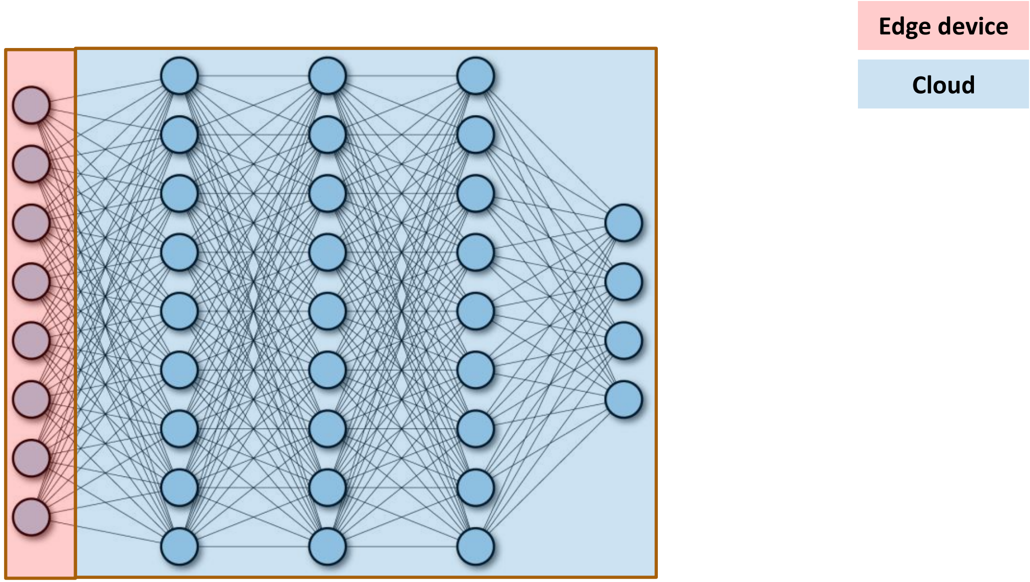


Towards data science

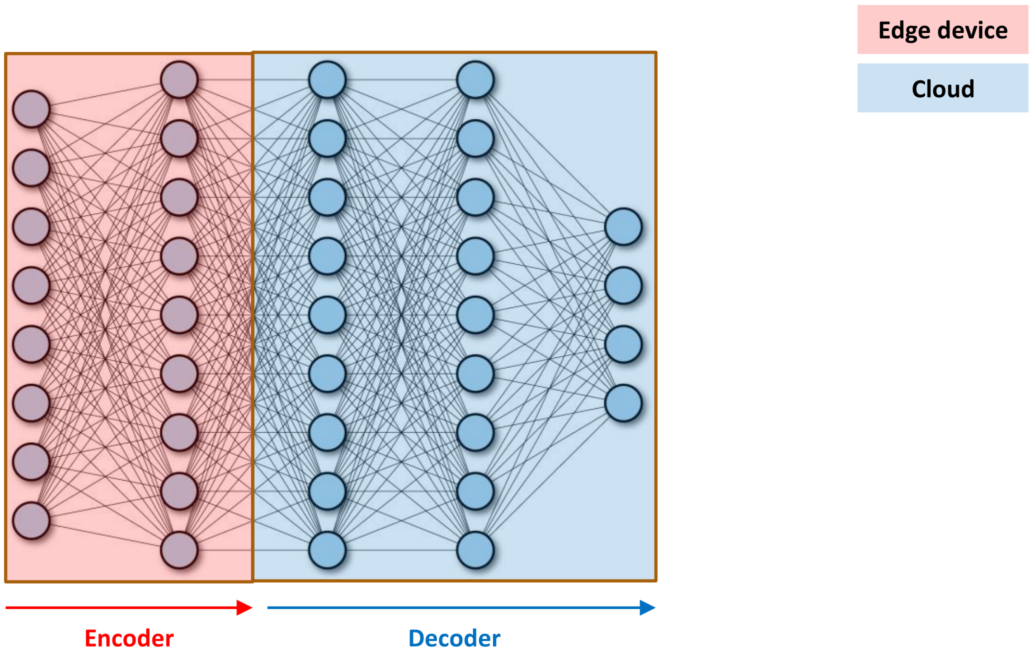
Fully connected

Convolutional

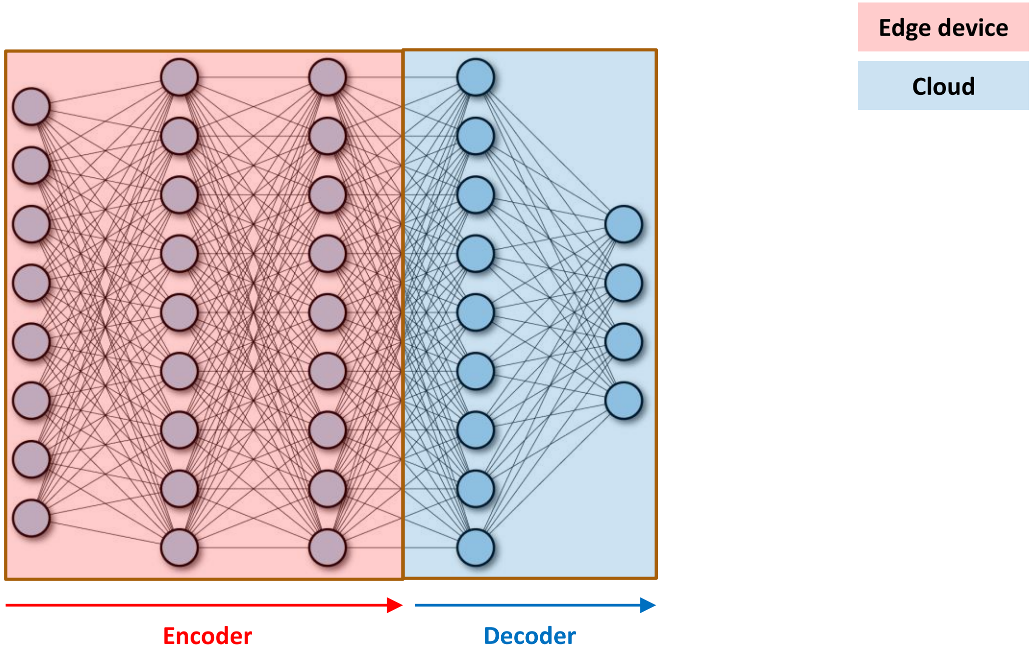
Distributed inference



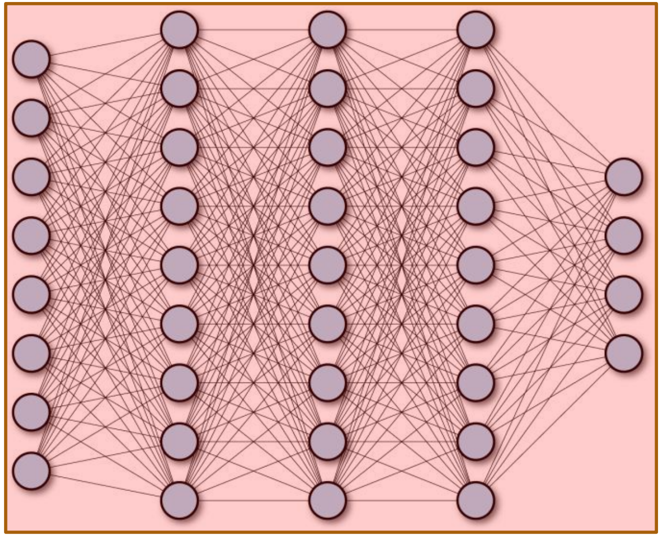
Distributed inference



Distributed inference



On-edge inference



Edge device

Cloud

DNN model compression

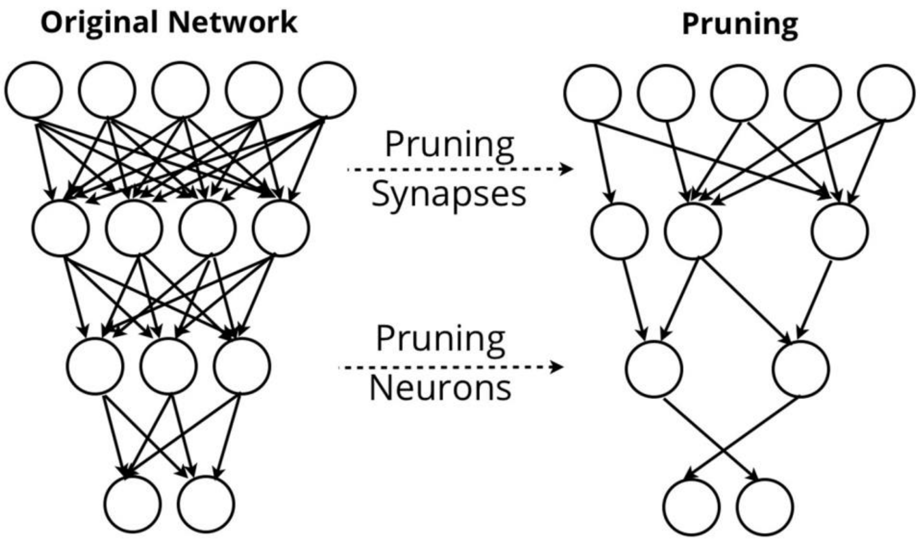
Goals:

- Reduce the implementation cost of DNNs
- Maintain the inference accuracy

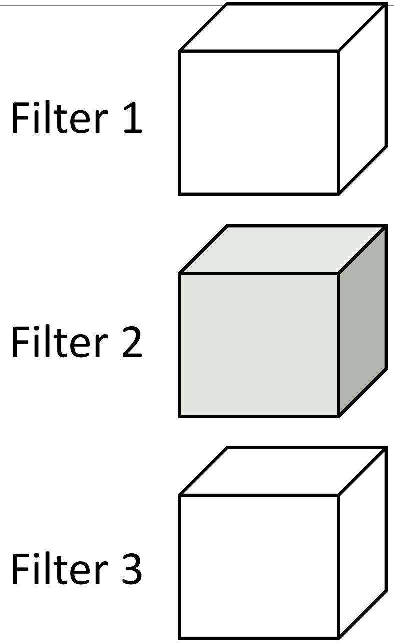
Methods:

- Weight pruning: structured or non-structured
- Weight quantization: leverages the inherent redundancy in the number of bits for weight representation

Weight pruning



Non-structured



Structured

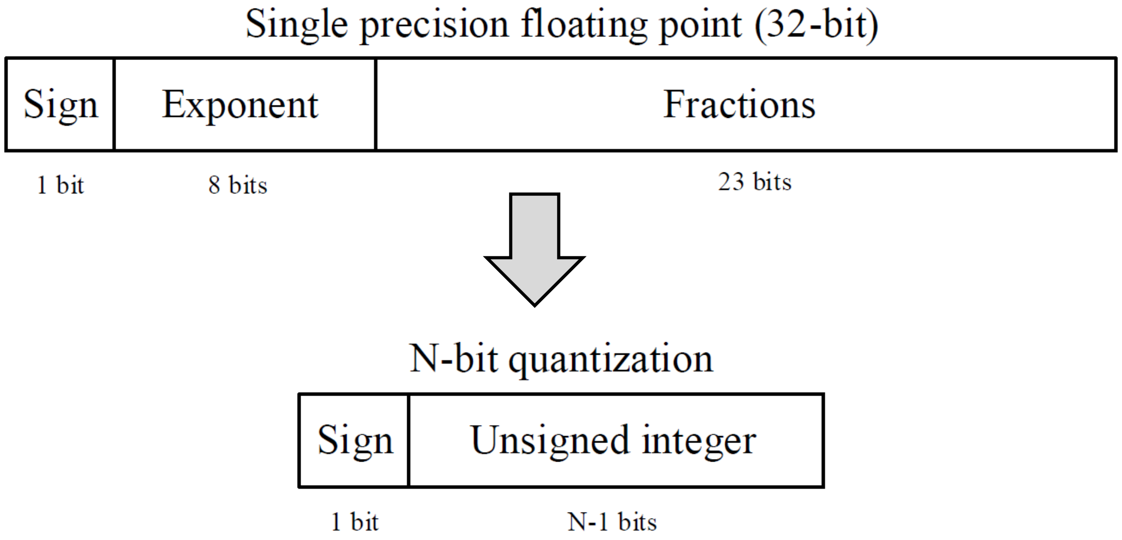
Weight quantization

Goals:

- Facilitate hardware implementations
- Acceptable accuracy loss

Methods:

- Binary
- Ternary
- N-bit quantization



Faults

activation **propagation**

Cause-effect relationship **Cause-effect relationship**

Fault **Error** **Failure**

Error Latency

A physical defect that occurs within HW or SW components:
- Defect for HW tech
- Bug for SW tech

A deviation from accuracy or correctness:
- The manifestation of a fault

The nonperformance of some action that is due or expected:
- Malfunction

Physical Informational External

Low-level **High-level**

Storage faults

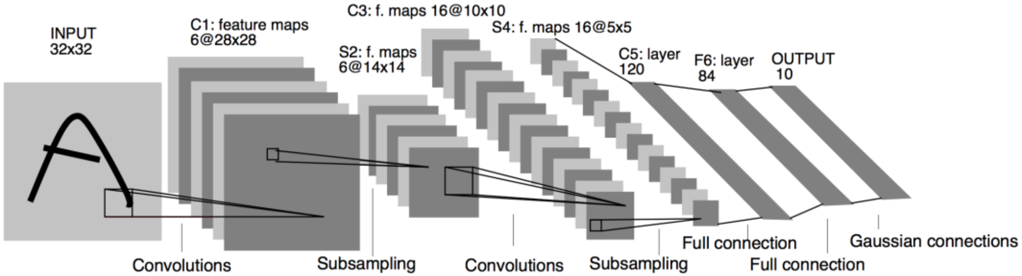
Defined as: any alteration in the intended storage state which affects the execution of a program on an otherwise functional unit

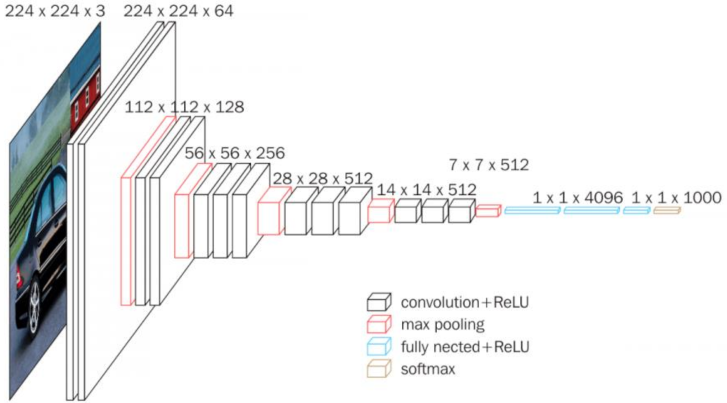
Sources:

- External: electromagnetic, laser, voltage-frequency based attacks
- Internal: local or remote adversary creating unstable or faulty condition

Experimental Setup

Evaluating Fault resiliency of compressed and uncompressed models of LeNet-5 and VGG16





LeNet-5
3 CONV layers
2 FC layers

VGG16
12 CONV layers
4 FC layers

The resiliency metric and experimental steps

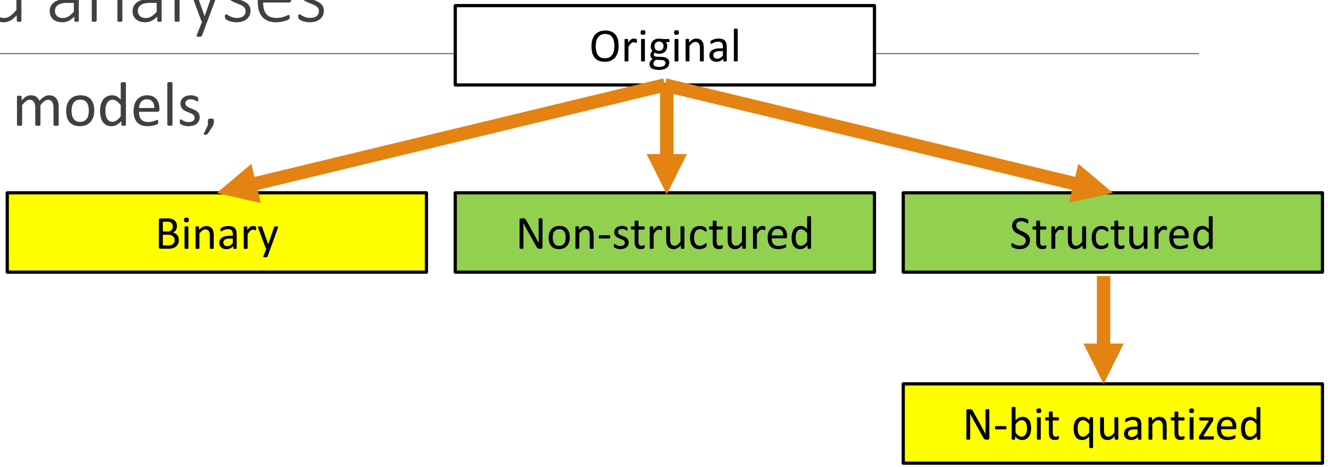
Maximum bit error rate (BER) with zero accuracy degradation (BERZAD)

Steps:

1) Load the target model (compressed or uncompressed)
2) Select target layers (convolution vs. fully-connected)
3) Based on BER, randomly select some bits from the weights of that layer and flip them
4) Store back the weights and test the inference accuracy of faulty models
5) Repeat 1-4 for ten times and record the average (prediction) accuracy for each model

Models and analyses

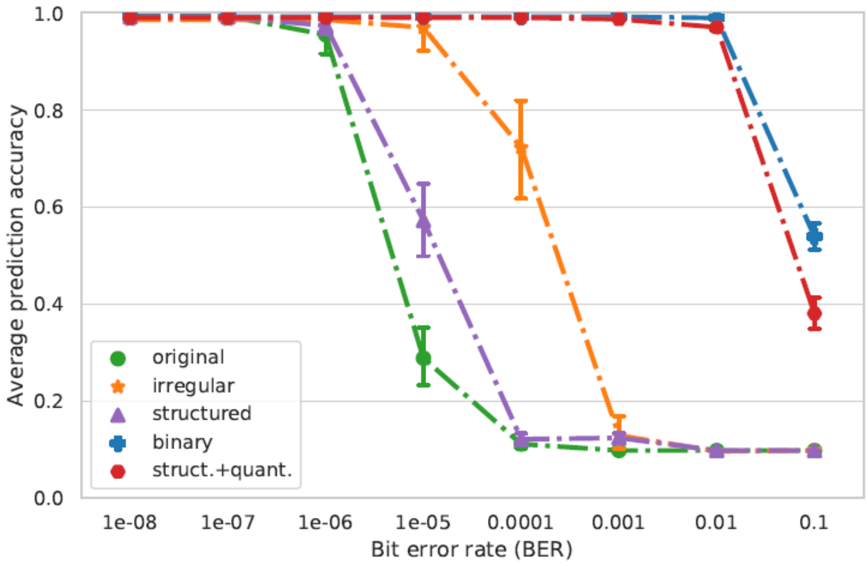
We consider 5 models,



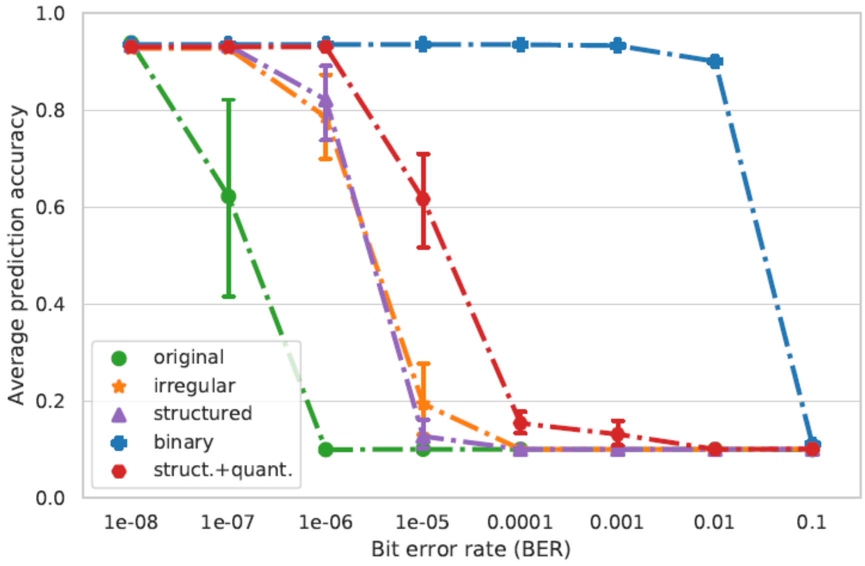
And two types of prediction accuracy analyses for each:

- Overall accuracy analysis (different data types and compression methods)
- Per-layer accuracy analysis (network layer types)

Results: overall accuracy analysis

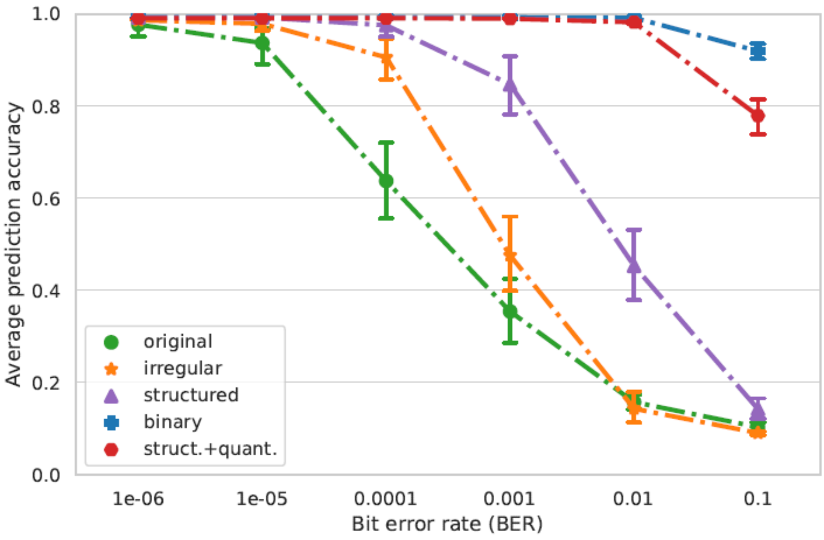


LeNet-5

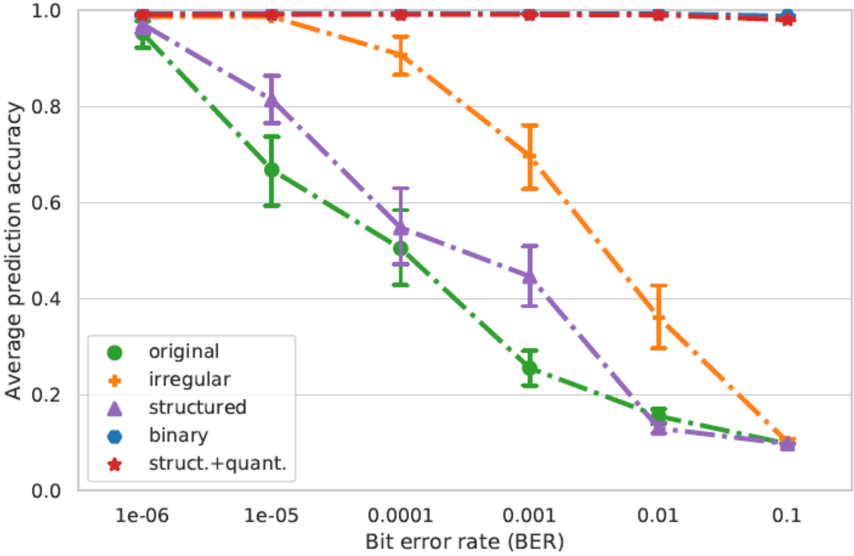


VGG16

Results: per-layer accuracy analysis – LeNet-5

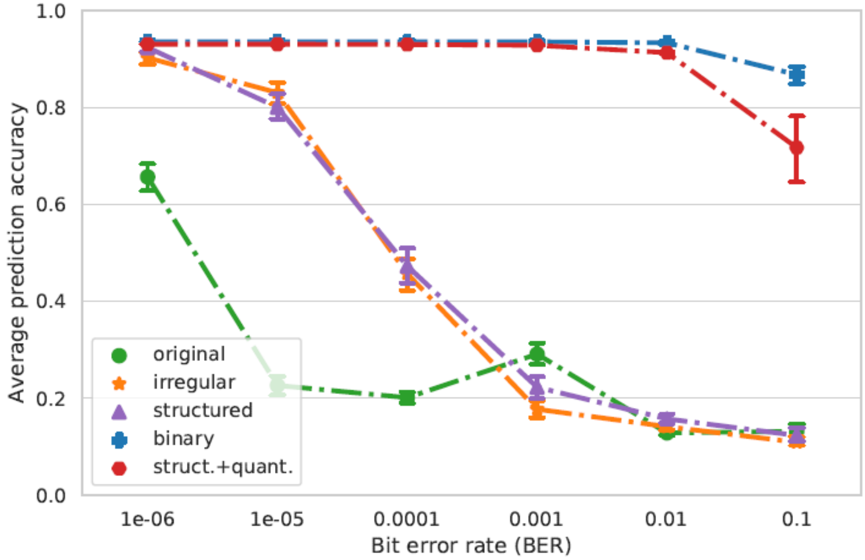


Convolutional layers

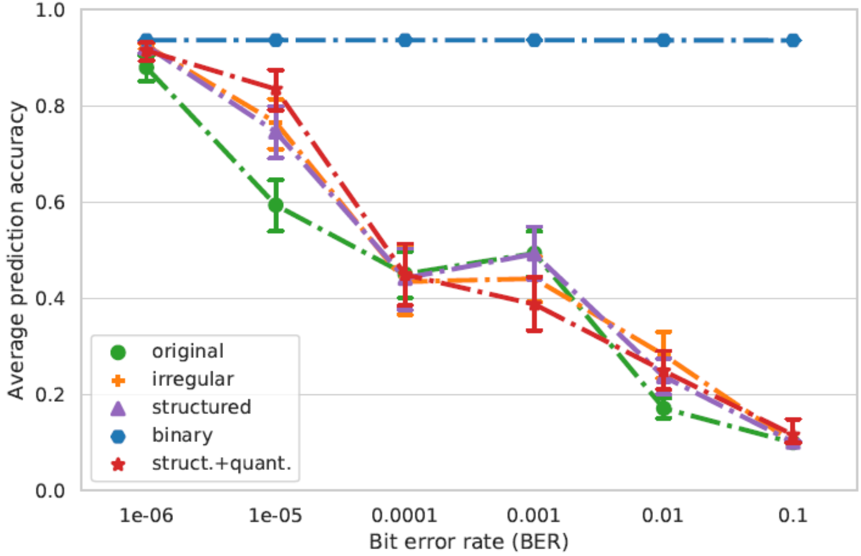


Fully-connected

Results: per-layer accuracy analysis – VGG16



Convolutional layers



Fully-connected

Sparsity effect

DNN name	Compression Type	Total Sparsity	BERZAD
LeNet-5	Original (uncompressed)	0%	10^{-7}
	Irregularly pruned (non-structured)	95.58%	10^{-7}
	Structured pruned	5.77%	10^{-7}
	Binary quantized	0%	10^{-3}
	Structured pruned + 3-bit quantized	43.73%	10^{-4}
VGG16	Original (uncompressed)	0%	10^{-8}
	Irregularly pruned (non-structured)	93.62%	10^{-7}
	Structured pruned	94.34%	10^{-7}
	Binary quantized	0%	10^{-4}
	Structured pruned + 5-bit quantized	94.82%	10^{-6}

Implementing Fault resilient DNNs

For hardware, quantization is the key!

- Binary quantization outperforms others in terms of implementation cost and fault resiliency
- Quantize all layers!

For software:

- Structured pruning for the convolutional layers + irregular pruning for fully-connected layers

Mitigation techniques

Even the DNN systems with compressed models are vulnerable to fault attacks → protection mechanisms is needed

Common approaches:

- Design of resilient DNN architectures
- Software-based techniques
- Hardware-based techniques

Our solution:

- Performing intermediate health checks during inference
- Adding duplicate hardware
- Enabling ECC on all large memory

Future work

- Evaluate the resiliency of other DNNs models
- Study models with different compression techniques
- Injecting faults on real platforms (via EM, laser, etc.) and comparing the experimental data with the field observations

Conclusion

- We developed a simulation framework for assessing the resiliency of DNN models against weight faults.
- Compressed models are more fault resilient compared to uncompressed models
- Proper weight quantization significantly enhances the resiliency of the DNN
- Binary quantized models exhibit the best resiliency against fault attacks, while being an ideal choice for embedded deployment

Thank you!

Comments/Questions?

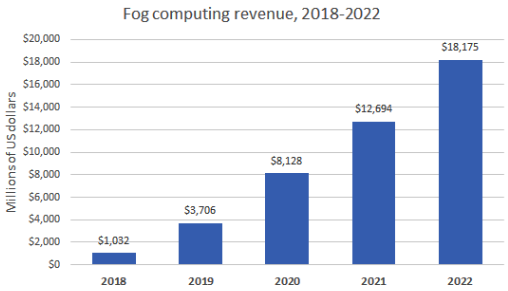
Majid Sabbagh

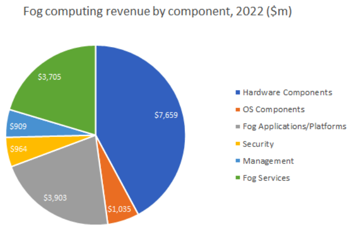
sabbagh.m@husky.neu.edu

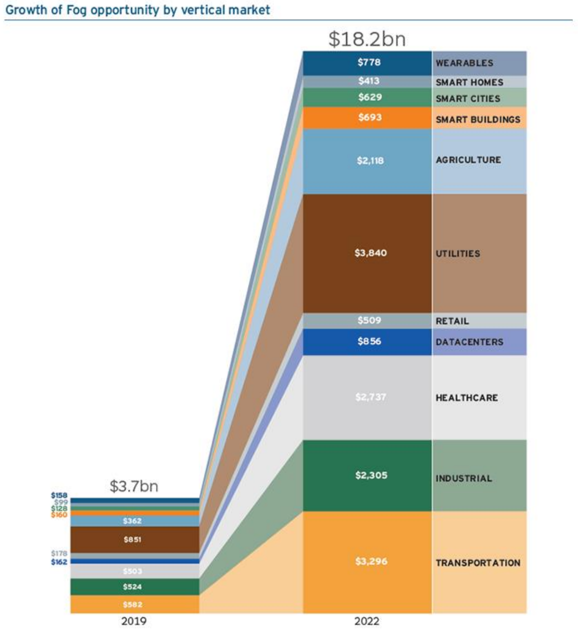
Edge computing market trends

Edge computing market will be worth $6.72 billion by 2022, with a CAGR (Compound Annual Growth Rate) of 35.4 percent (MarketsandMarkets analysts)

Fog computing revenue







Data: 451 Research & OpenFog Consortium / Chart: ZDNet

Image: 451 Research & OpenFog Consortium