Twitter Thread by Lea Kissner



Lea Kissner
@LeaKissner

Next up at #enigma2021, Sanghyun Hong will be speaking about "A SOUND MIND IN A VULNERABLE BODY: PRACTICAL HARDWARE ATTACKS ON DEEP LEARNING"

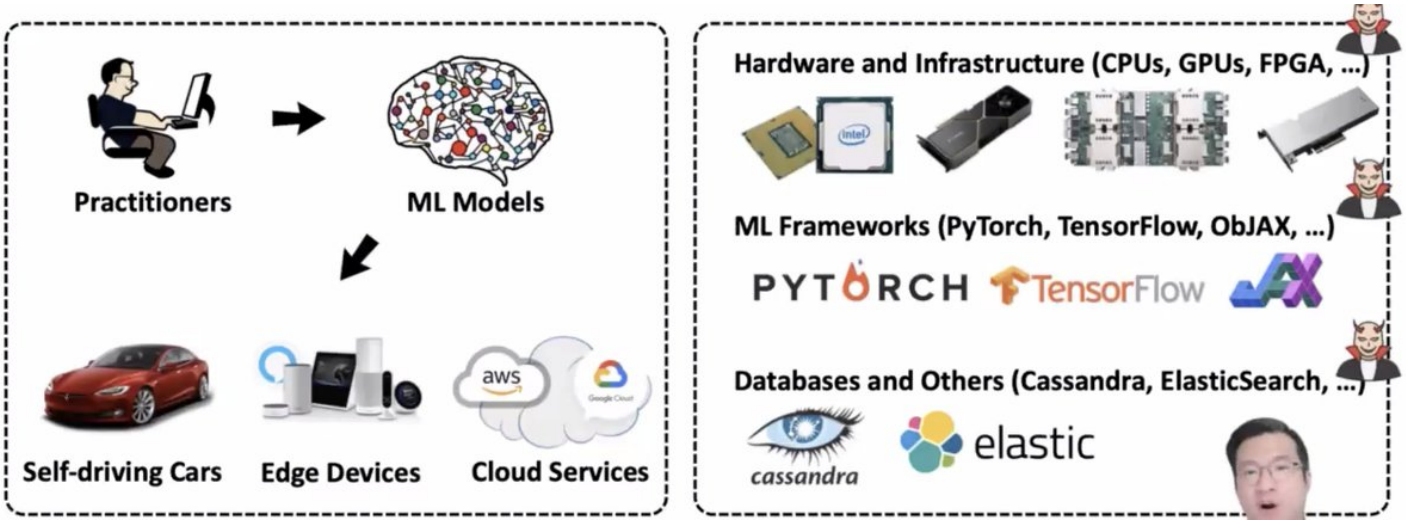
(Hint: speaker is on the

In recent years ML models have worked from research labs to production, which makes ML security important. Adversarial ML research studies how to mess with ML

For example by messing with the training data (c.f. Tay which became super-racist super-fast) or by foiling ML models by changing inputs in ways humans can't see.



Prior work considers ML models in a standalone, mathematical way
* looks at the robustness in an isolated manner
* doesn't look at the whole ecosystem and how the model is used -- ML models are running in real hardware with real software which has real vulns!



This talk focuses on hardware-level vulnerabilities. This is particularly interesting because these can break cryptographic guarantees (because those are outside of their threat models)
e.g. fault injection attacks, side-channel attacks

Recent work targets The Cloud
* co-location of VMs from different users
* weak attackers with less subtle control

The cloud providers try to secure things, e.g. protections against Rowhammer

But can you use the weak attacks left after mitigations deployed by cloud compute providers?

DNNs are resilient to numerical perturbations: this is used both to make things more efficient (e.g. pruning) but also in security it's really hard to make accuracy drop

... BUT this focuses on the average or best case, not the worst cast!

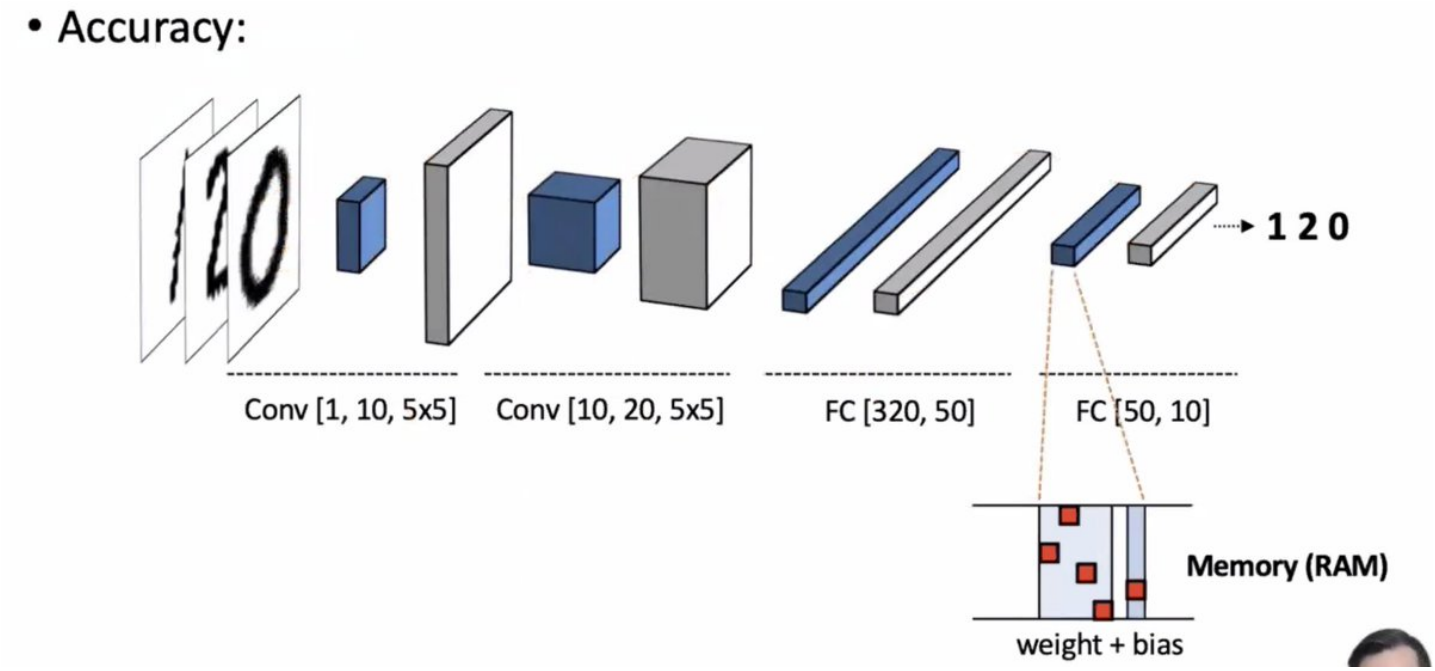
DNNs Are Known to Be Resilient to Numerical Perturbations

- **Techniques** that rely on the *graceful degradation*
  - **Parameter pruning**[1]: to reduce the inference cost
  - **Parameter quantization**[2]: to compress the network size
  - **Blend noises to parameters**[3]: to improve the robustness

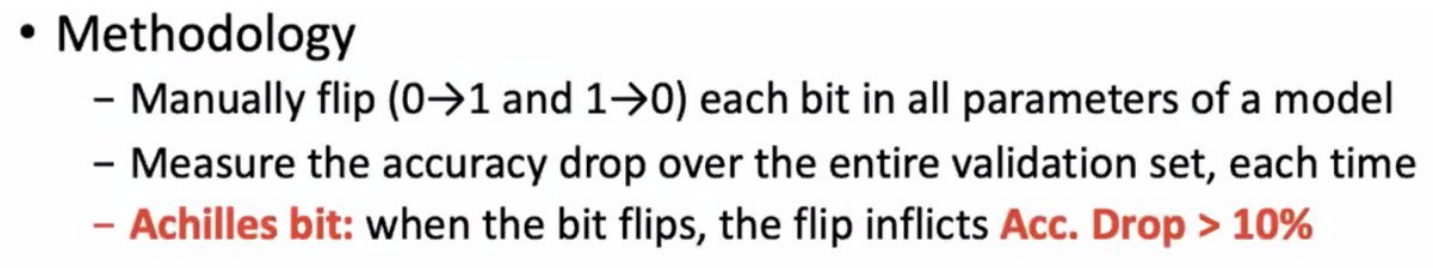
- **Prior work** showed it is difficult to cause the accuracy drop
  - **Indiscriminate poisoning**[4]: blend a lot of poisons ≈ ***11% drop***
  - **Storage media errors**[5]: a lot of random bit errors ≈ *5% drop*
  - **Hardware fault attacks**[6,7]: a lot of random faults ≈ *7% drops*

What happens when you can mess with the memory at one of these steps?
* negligible effect on the average case accuracy
* but flipping one bit can make significant amount of damage for particular queries

How much damage can a single bit flip cause?



Try it out!

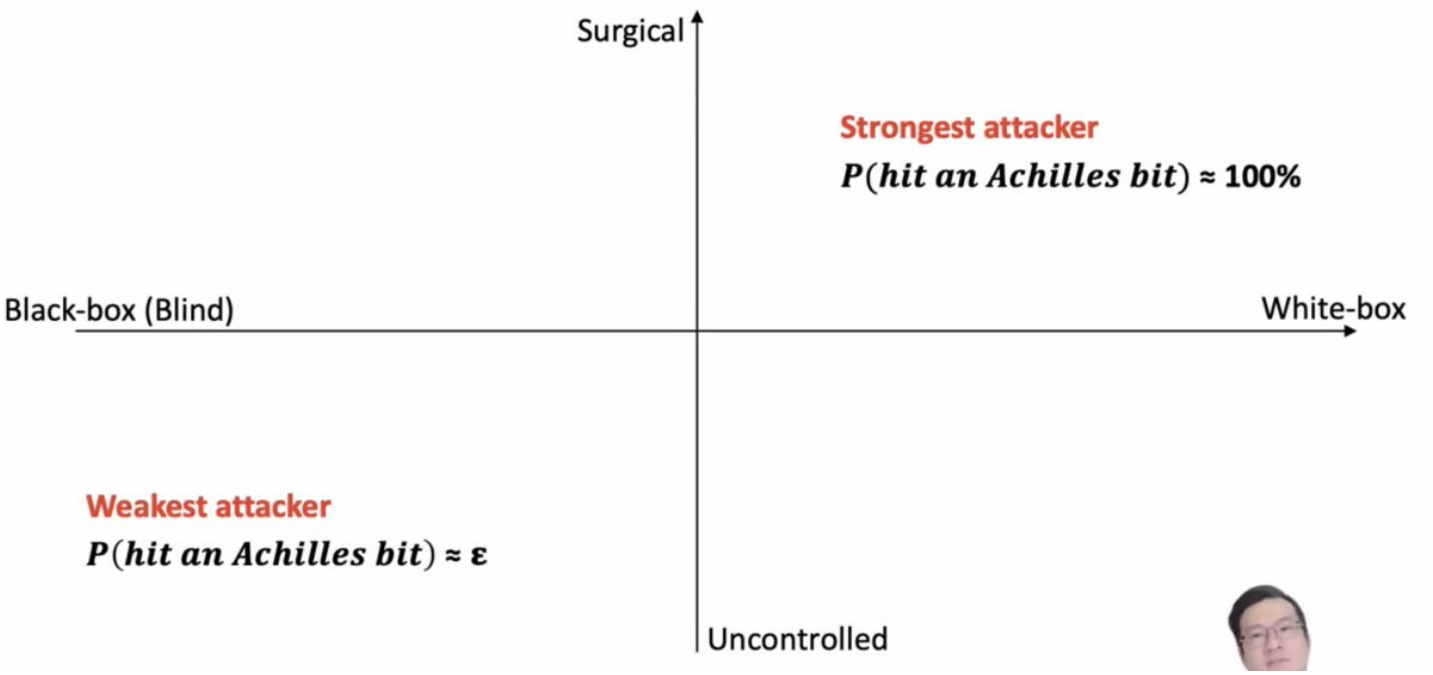


tl;dr in general, one bit flip can really mess with your model! (Looked for the worst bit to flip)

Dataset	Network	Acc.	# Params	Acc. Drop	Ratio
CIFAR-10	B(ase)	83.74	776K	94 %	46.8%
	B-Slim	82.19	197K	93 %	46.7%
	B-Dropout	81.18	776K	94 %	40.5%
	B-D-Norm	80.17	778K	97 %	45.9%
	AlexNet	83.96	2.5M	96 %	47.3%
	VGG16	91.34	14.7M	99 %	46.2%
ImageNet	AlexNet	79.07	61.1M	100 %	47.3%
	VGG16	90.38	138.4M	99 %	42.1%
	ResNet50	92.86	25.6M	100 %	47.8%
	DenseNet161	93.56	28.9M	100 %	49.0%
	InceptionV3	88.65	27.2M	100 %	40.8%

Well, can you use this? There's a lot less control in real life

Some strong attackers might be able to hit an "achilles" bit (one that's really going to mess with the model), but weaker attackers are going to hit bits more randomly.



So they tried it out!

- **Setup:**
  - **MLaaS scenario:** run a VM under the Rowhammer pressure
  - **Rowhammer attack:**
    - Random bit-flip: one can **flip bits in either the code or data** region in memory
    - Run our experiments on **12 different DRAM chips**
    - Run **25** experiments for each of 12 DRAM chips
    - Run **300** cumulative bit-flip attempts for each experiment

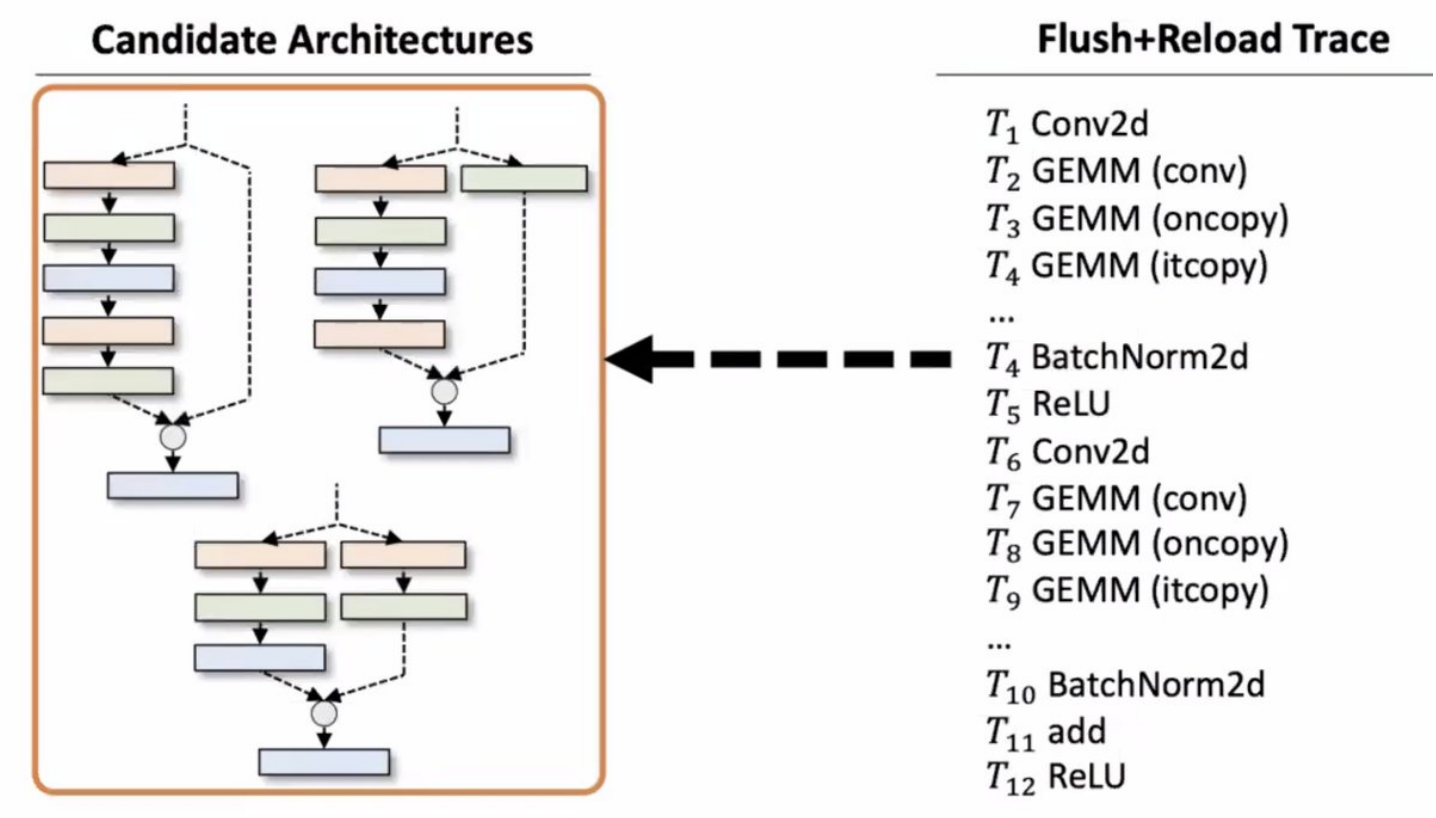
tl;dr running a pretty weak Rowhammer attack is enough to mess with a ML model being trained.

- **Summary of our results:**
  - On average, **62%** of our experiments causes **an accuracy drop > 10%**
  - In successful attacks, **the time** it takes to drop the accuracy is **< few minutes**
  - Our attacks are **inconspicuous**: only 6 program crashes (0.08%) over 7.5 bit-flips

How about side-channel attacks?

The attacker might want to get their hands on fancy DNNs which are considered trade secrets and proprietary to their creators. They're expensive to make! They need good training data! People want to protect them!

Prior work required that the ML-model-trainer uses an off-the-shelf architecture. But people often don't for the fancy models. So what this work does [... if I'm following correctly] is to basically guess from a lot of architecture possibilities and then filter it down



Why is this possible? Because there are regularities in deep-learning calculation.

Does this work? Apparently so: they tried it out using a cache side-channel attack and got back the architectures of the fancy DNN back.

- **Setup:**
  - **MLaaS scenario:** run a VM monitored by Flush+Reload attacker
  - **Unique DNNs:** MalConv and ProxylessNAS-CPU

- **Summary of our result:**
  - Our attacker reconstructed both unique DNN architectures **with 0 error**
  - The time it takes to steal those architectures:
    - **MalConv:** **a few minutes**
    - **ProxylessNAS-CPU:** **12 CPU hours** < 40k GPU hours for running NAS

This needs more study
* we need to understand the worst-case ML fails under hardware attack
* don't discount the ability of an attacker with access to a weak hardware attack to cause a disproportionate amount of damage

You can find a writeup of this research at https://t.co/qUx8nAHW52

[end of talk]