Sponge Examples: Energy-Latency Attacks on Neural Networks

Ilia Shumailov*, Yiren Zhao*, Daniel Bates*, Nicolas Papernot^, Robert Mullins*, Ross Anderson*

* University of Cambridge
^ University of Toronto, Vector Institute
Machine Learning

- Machine learning is everywhere
- We operate based on data, not formal rules
- There’s a lot of non-determinism
- It is suddenly hard to define Security

https://xkcd.com/1838/
Computer Security in context of Machine Learning

- Adversarial examples exist for all models
- A large taxonomy of attackers
- Work in White / Grey / Black-box settings
- Attacks are scalable because of transferability
Machine Learning in context of Computer Security

- ML is a part of a larger pipeline
- As secure as the weakest component
- Clear threat model
- Safety and Security policies and cases
- Existence of trusted components
- Well defined environment
Machine Learning in context of Computer Security

[D] Possible malware found hidden inside images from the ImageNet dataset

Vulnerability Details: CVE-2018-8825

Google TensorFlow 1.7 and below is affected by: Buffer Overflow. The impact is: execute arbitrary code (local).

Publish Date: 2019-04-23 Last Update Date: 2019-04-25

TensorFlow models are programs

TensorFlow’s runtime system interprets and executes programs. These programs that TensorFlow executes. TensorFlow programs are executed separately in checkpoints.

At runtime, TensorFlow executes the computation graph using the data that may change depending on the parameters provided. TensorFlow may read and write files, send and receive data over the network, and it is performed with the permissions of the TensorFlow process. Allow
Machine Learning in context of Computer Security

system, hazard, risk, error, failure, threat, accident, safety case, security policy, trust, reliability, subject, person, principal, secrecy, privacy, confidentiality, anonymity, integrity, availability, authenticity, uncertainty, and safety
Machine Learning in context of Computer Security

system, hazard, risk, error, failure, threat, accident, safety case, security policy, trust, reliability, subject, person, principal, secrecy, privacy, confidentiality, anonymity, integrity, availability, authenticity, uncertainty, and safety

**Safety** looks at average case, **Security** considers worst case

*What is a worst case for an ML component?*
Availability

Ensuring **timely** and **reliable** access to and use of information. (NIST Special Publication 800-12)
Availability

Benign Data

Sponge Examples

Increased latency

Over-heating and over-consumption of energy
Energy Gap

The amount of energy consumed by one inference pass (i.e. a forward pass in a neural network) depends primarily on:

- The overall **number of arithmetic operations** required to process the inputs;
- The **number of memory accesses** e.g. to the GPU DRAM.
Hypothesis 1: Data Sparsity

Optimisations exploit runtime **data sparsity** to increase efficiency.

- Zero-skipping multiplications;
- Encoding DRAM traffic to reduce the off-chip bandwidth requirement.
Hypothesis 2: Computation Dimensions

Modern networks have a **computational dimension**

- A large number of NLP models are **auto-regressive** e.g. RNNs and GPT2
- **Adaptive** input **dimensions** to help performance e.g. GPT2 uses Byte Pair Encoding
- ML components are a **part of loop**
Hypothesis 2: Computation Dimensions for GPT2

**Auto-regressiveness** adds an unbounded loop

**Algorithm 1**: Translation Transformer NLP pipeline

Result: $y$

1. $\downarrow O(l_{\text{tin}})$
2. $x_{\text{tin}} = \text{Tokenize}(x)$;
3. $y_{\text{tout}} = \emptyset$
4. $\downarrow O(l_{\text{ein}})$
5. $x_{\text{ein}} = \text{Encode}(x_{\text{tin}})$;
6. $\downarrow O(l_{\text{tin}} \times l_{\text{ein}} \times l_{\text{tout}} \times l_{\text{eout}})$
7. while $y_{\text{tout}}$ has no end of sentence token do
   8. $\downarrow O(l_{\text{eout}})$
   9. $y_{\text{eout}} = \text{Encode}(y_{\text{tout}})$;
   10. $\downarrow O(l_{\text{ein}} \times l_{\text{eout}})$
   11. $y_{\text{eout}} = \text{model.
Inference}(x_{\text{ein}}, y_{\text{eout}}, y_{\text{tout}})$;
   12. $\downarrow O(l_{\text{eout}})$;
   13. $y_{\text{tout}} = \text{Decode}(y_{\text{eout}})$;
   14. $y_{\text{tout}}.\text{add}(y_{\text{tout}})$;
8. end
16. $\downarrow O(l_{\text{tout}})$;
17. $y = \text{Detokenize}(y_{\text{tout}})$

**Encoding** adds **variable** I/O representation

**Benign with 4 tokens for input of size 16:**
Athazagoraphobia => ath, az, agor, aphobia

**1 error with 7 tokens for input of size 16:**
Athazagoraphobia => ath, az, agor, aph, p, bi, a

**Malicious with 16 tokens for input of size 16:**
A/h/z/g/r/p/p/i/ => A, /, h, /, z, /, g, /, r, /, p, /, p, /, i, /
Example of Computation Dimensions in Cognitive Radar

Block diagram of cognitive radar viewed as a dynamic closed-loop feedback system from *Cognitive radar: a way of the future*, Simon Haykin (2006)
Multiple ways to search for Sponge examples

- **White-box** gradient-based \(- \sum_{a_i \in A} \|a_i\|_2\) i.e. large activation norms across all hidden layers

- **Interactive White-box, Grey-box** and **Black-box** genetic algorithm-based
  - Perform inference on a sample
  - Measure energy consumed or inference time
  - Combine worst performing samples
  - Mutate
  - Repeat

- **Blind Black-box** attack genetic algorithm-based
  - Pick model solving similar task or using similar dictionary
  - Perform transferability attack
## White-box attack performance with NLP benchmarks

<table>
<thead>
<tr>
<th></th>
<th>Input size</th>
<th>NVML&lt;sub&gt;gpu&lt;/sub&gt;</th>
<th>Natural&lt;sub&gt;asic&lt;/sub&gt;</th>
<th>Random&lt;sub&gt;asic&lt;/sub&gt;</th>
<th>Sponge Mean&lt;sub&gt;asic&lt;/sub&gt;</th>
<th>Sponge Top 10%&lt;sub&gt;asic&lt;/sub&gt;</th>
<th>Energy&lt;sub&gt;gpu&lt;/sub&gt;</th>
<th>Time&lt;sub&gt;gpu&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Language Understanding: SuperGLUE Benchmark with [37]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CoLA</td>
<td>15</td>
<td>5829.32</td>
<td>4.30</td>
<td>69.72</td>
<td>83.92</td>
<td>87.11</td>
<td>×20.25</td>
<td>×1.23</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>9388.40</td>
<td>4.30</td>
<td>138.07</td>
<td>164.07</td>
<td>169.91</td>
<td>×39.51</td>
<td>×1.48</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>22698.87</td>
<td>4.30</td>
<td>452.49</td>
<td>518.19</td>
<td>530.80</td>
<td>×123.42</td>
<td>×3.82</td>
</tr>
<tr>
<td>MNLI</td>
<td>15</td>
<td>6126.65</td>
<td>12.88</td>
<td>73.47</td>
<td>86.97</td>
<td>89.96</td>
<td>×6.98</td>
<td>×1.32</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>9631.68</td>
<td>17.66</td>
<td>142.63</td>
<td>168.96</td>
<td>174.34</td>
<td>×9.87</td>
<td>×2.03</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>22952.14</td>
<td>34.47</td>
<td>456.11</td>
<td>518.89</td>
<td>531.40</td>
<td>×15.42</td>
<td>×3.16</td>
</tr>
<tr>
<td>WSC</td>
<td>15</td>
<td>27876.53</td>
<td>14.48</td>
<td>523.28</td>
<td>1300.19</td>
<td>2152.67</td>
<td>×148.62</td>
<td>×9.83</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>82822.58</td>
<td>34.94</td>
<td>1882.63</td>
<td>3927.63</td>
<td>5348.06</td>
<td>×153.08</td>
<td>×19.25</td>
</tr>
<tr>
<td></td>
<td>100</td>
<td>662811.96</td>
<td>194.89</td>
<td>16754.13</td>
<td>25367.30</td>
<td>30692.95</td>
<td>×157.49</td>
<td>×69.83</td>
</tr>
<tr>
<td><strong>Machine Translation: WMT14/16 with [41]</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>En→Fr</td>
<td>30</td>
<td>59597.32</td>
<td>31.87</td>
<td>109.80</td>
<td>118.47</td>
<td>141.27</td>
<td>×4.43</td>
<td>×4.45</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>93731.34</td>
<td>48.54</td>
<td>166.13</td>
<td>249.89</td>
<td>569.85</td>
<td>×11.74</td>
<td>×13.51</td>
</tr>
<tr>
<td>En→De</td>
<td>15</td>
<td>18133.66</td>
<td>18.19</td>
<td>35.80</td>
<td>242.39</td>
<td>542.35</td>
<td>×29.82</td>
<td>×32.86</td>
</tr>
</tbody>
</table>

Energy is reported in millijoules. GA was ran for 100 epochs with a pool size of 100.
## White-box attack performance for CV tasks

<table>
<thead>
<tr>
<th></th>
<th>Time_\text{gpu} [s]</th>
<th>Cost_\text{asic} [mJ]</th>
<th>Cost_\text{asic ratio}</th>
<th>post-ReLU Density</th>
<th>Density</th>
<th>Max Density</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ResNet-50</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L-BFGS-B Sponge</td>
<td>0.011</td>
<td>164.727</td>
<td>0.863</td>
<td>0.619</td>
<td>0.885</td>
<td></td>
</tr>
<tr>
<td>Sponge</td>
<td>0.016</td>
<td>160.887</td>
<td>0.843</td>
<td>0.562</td>
<td>0.868</td>
<td></td>
</tr>
<tr>
<td>Natural</td>
<td>0.017</td>
<td>160.562</td>
<td>0.842</td>
<td>0.572</td>
<td>0.867</td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td>0.017</td>
<td>155.820</td>
<td>0.817</td>
<td>0.483</td>
<td>0.845</td>
<td></td>
</tr>
<tr>
<td><strong>DenseNet-121</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L-BFGS-B Sponge</td>
<td>0.033</td>
<td>152.595</td>
<td>0.783</td>
<td>0.571</td>
<td>0.826</td>
<td>0.998</td>
</tr>
<tr>
<td>Sponge</td>
<td>0.029</td>
<td>149.564</td>
<td>0.767</td>
<td>0.540</td>
<td>0.814</td>
<td></td>
</tr>
<tr>
<td>Natural</td>
<td>0.033</td>
<td>147.227</td>
<td>0.755</td>
<td>0.523</td>
<td>0.804</td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td>0.030</td>
<td>144.365</td>
<td>0.741</td>
<td>0.487</td>
<td>0.792</td>
<td></td>
</tr>
<tr>
<td><strong>MobileNet v2</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>L-BFGS-B Sponge</td>
<td>0.011</td>
<td>87.511</td>
<td>0.844</td>
<td>0.692</td>
<td>0.890</td>
<td>0.996</td>
</tr>
<tr>
<td>Sponge</td>
<td>0.010</td>
<td>84.513</td>
<td>0.815</td>
<td>0.645</td>
<td>0.868</td>
<td></td>
</tr>
<tr>
<td>Natural</td>
<td>0.011</td>
<td>85.075</td>
<td>0.821</td>
<td>0.646</td>
<td>0.873</td>
<td></td>
</tr>
<tr>
<td>Random</td>
<td>0.011</td>
<td>80.805</td>
<td>0.779</td>
<td>0.567</td>
<td>0.844</td>
<td></td>
</tr>
</tbody>
</table>

Energy is reported in millijoules. GA was ran for 100 epochs with a pool size of 100.
Interactive Black-box attack performance against WMT16 En→Fr benchmark

Figure 1: Performance of Sponge Examples based on the Energy, Time and Simulator fitness costs.
Do sponges exist in practice? Yup
Towards defences again Sponge Examples

- Lesson from Computer security: **optimisations increase attack surface**
  - Side channel attacks
  - Denial-of-service attacks

- Optimisations **widen** average to worst case **time-energy gap**

- Not clear how to keep performance and security
  - Still have not solved Spectre & Meltdown
  - Constant time computation solves security issues, but things get too slow

- **Potential simple defense:**
  - Kill inference when more than average amount of time or energy is consumed
  - Will cause a lot of false positives and make jamming easy. Can we do better?

- **Real-time systems** in presence of Sponges
  - Can Tesla collision avoidance system afford to not make a decision?
  - What should be the maximum energy gap for RT?
Conclusions

- It is possible to attack model **availability** in both White and Black-box settings
- Attack can target **hardware optimisations**
  - For some CV tasks we fully negated benefits from acceleration
- Attacks can target **algorithmic complexity**
  - For some NLP tasks we managed to get up to x200 energy consumption and x70 time
- Average case is very different from worst case scenario
- **Impact** of ML on climate change might have been **underestimated**
- It is **not clear how to defend** systems against Sponge examples
- **Real-time systems** with ML components **should model availability** adversary
Thank you very much for listening!

Please do not hesitate to reach out in case there are any questions at

ilia.shumailov@cl.cam.ac.uk