Modeling the Resource Requirements of Convolutional Neural Networks on Mobile Devices

Zongqing Lu, Swati Rallapalli, Kevin Chan, and Thomas La Porta

Abstract: Convolutional Neural Networks (CNNs) have revolutionalized the research in computer vision. We envision CNNs will be eventually and widely deployed on mobile devices. Therefore, we aim to understand the resource requirements (time, memory) of CNNs on mobile devices.

- We measure and analyze the performance and resource usage on a layer-wise granularity. *Our findings point out the potential ways of optimizing the performance on mobile devices.*
- We build *SiteCNN*, *which takes a CNN configuration as the input and estimates the compute time and resource usage of the CNN.*

Motivation: Whether and how efficiently does a CNN can be run on a given mobile platform? As CNNs vary from several layers to thousand layers, the answer can serve as guidelines to decide when performance optimizations, offloading, etc. are required to successfully run analytics tasks on mobile devices.

Challenges:
- Different types of layers
- Profiling overhead on mobile GPUs
- How matrix multiplciation scales

Measurement Set-Up

Platform: NVIDIA TK1 and TX1
- TK1: 2.3GHz quad-core Cortex-15A 32bit CPU
  192 CUDA cores Kepler GPU
- TX1: 1.9GHz quad-core Cortex-A57 64bit CPU
  256 CUDA cores Maxwell GPU

Framework: Caffe

CNNs: AlexNet, VGGNet, GoogleNet, ResNet

Measurement (Timing)

Table 1: Timing benchmark on VGGNet

Model	Platform		Layerwise Pass (ms)					Forward Pass (ms)
			CONV	POOL	ReLU	FC	Total	
VGGNet	TK1	CPU	7160.5±0.7 **93.02%**	60.1±0.1 0.78%	95.6±0.1 1.24%	381.6±0.2 4.96%	7697.9±0.6	7697.8±0.5
		GPU	263.1±19.3 **75.68%**	7.2±0.5 2.06%	17.5±1.2 5.03%	57.6±0.5 16.58%	347.6±20.1	326.7±2.1
	TX1	CPU	1952.9±12.2 **69.14%**	71.3±1.5 2.52%	52.5±1.9 1.86%	747.7±24.9 26.47%	2824.6±23.2	2809.1±10.6
		GPU	136.3±5.4 **73.98%**	3.4±1.6 1.84%	9.9±4.9 5.35%	32.8±1.3 17.82%	184.2±7.4	175.3±2.0
	FLOPs		15360M **99.08%**	6M 0.04%	14M 0.09%	124M 0.79%	15503M	

Measurement (Memory)

Table 2: Memory of CNNs on platforms (MB)

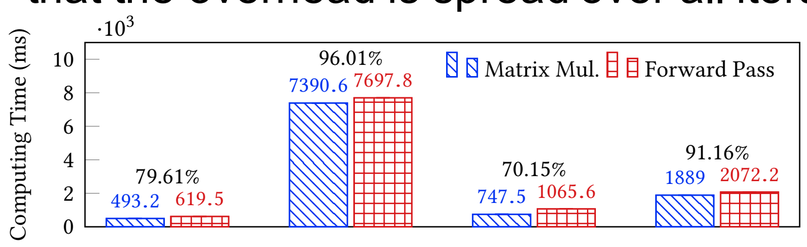
Type/Platform		AlexNet	VGGNet	GoogleNet	ResNet
Weights&Biases		233	528	26	97
Data		8	110	53	221
Workspace		11	168	46	79
TK1	CPU	324	972	161	409
	GPU	560	1508	196	533
TX1	CPU	362	1013	200	453
	GPU	589	1537	226	562

Our findings:
- Matrix multiplication takes the major proportion of computation, memory, and computing time of a CNN.
- Layerwise measurement overhead on GPUs is large.
- Some types of layers run faster on CPU than GPU.

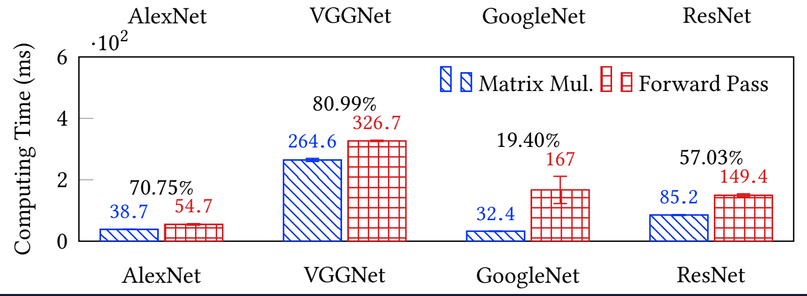
Potential performance optimization on mobile devices:
- Take advantage of united memory architecture to minimize memory usage (zero-copy).
- Perform each layer at the best locality (either CPU or GPU) for acceleration

Profiling

Strategies:
- Extract matrix multiplications of a CNN.
- Approximatethe computing by the underlying matrix multiplications.
- Mitigate profiling overhead by running a computing task iteratively without incuring synchronization such that the overhead is spread over all iterations.

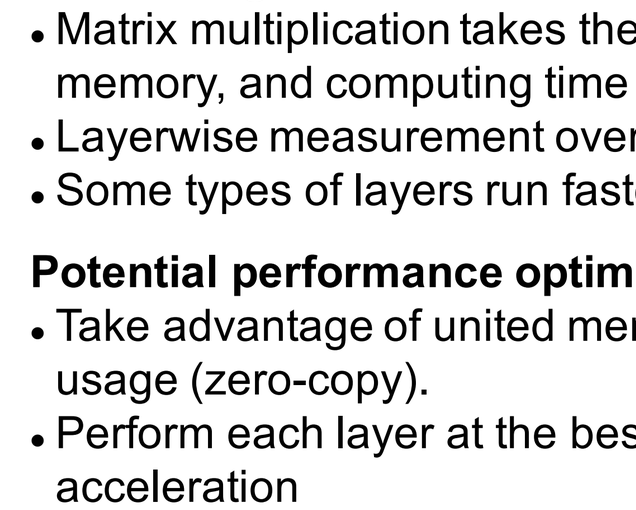


TK1 CPU

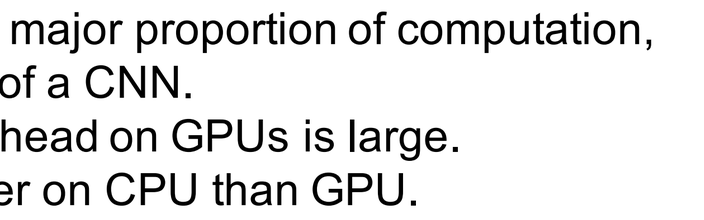


TK1 GPU

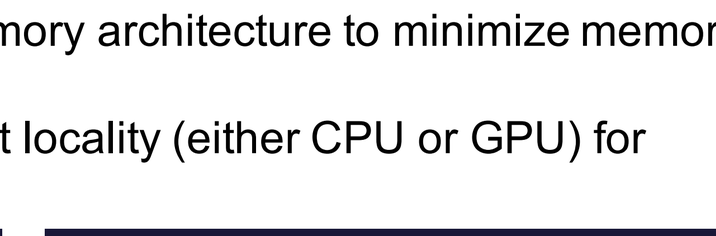
Modeling



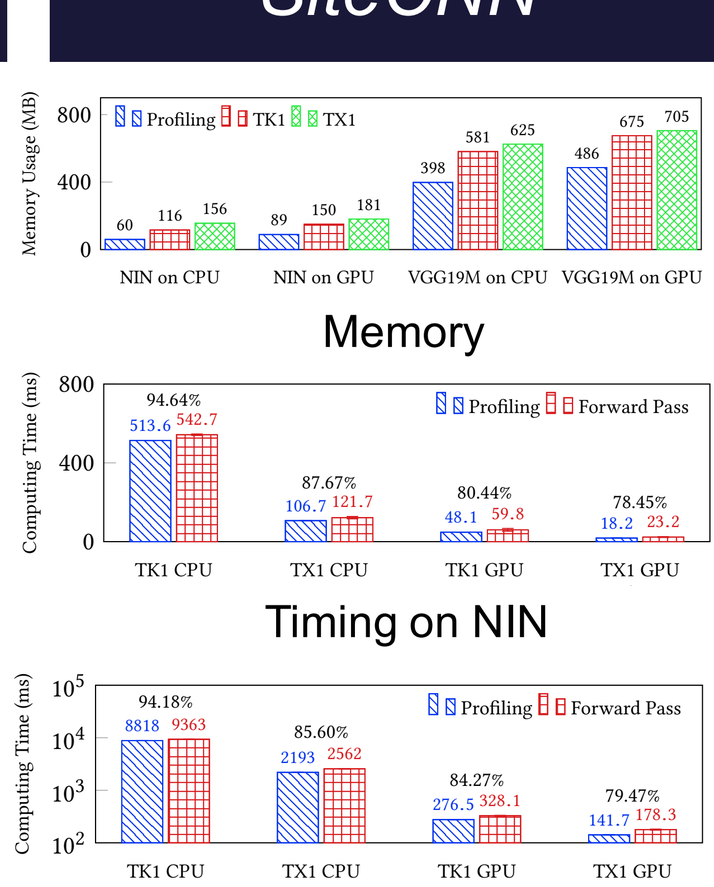
SiteCNN



Memory



Timing on NIN



Timing on VGG19