Institut für Computertechnik
Institute of Computer Technology

Performance Modeling of DNNs for Embedded Platforms

Eröffnung Josef Ressel Zentrum
Künstliche Intelligenz für ressourcenbegrenzte Geräte

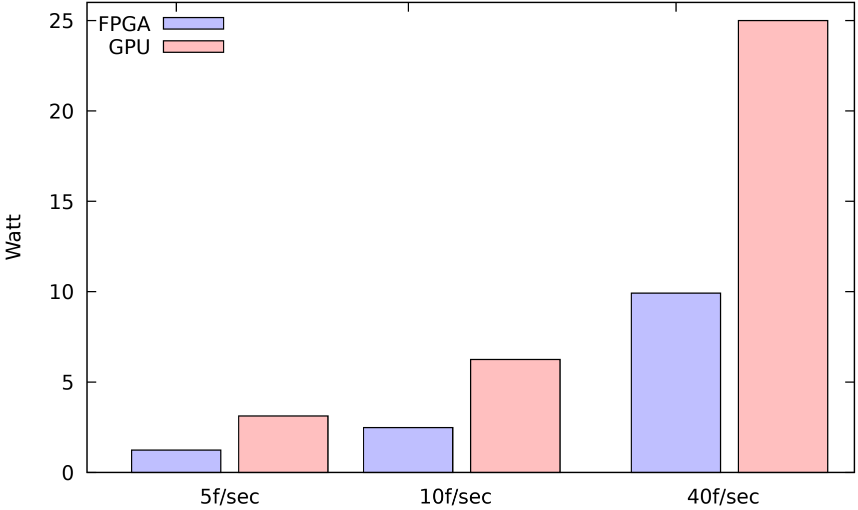
Axel Jantsch

Hagenberg, May 8, 2024

Outline

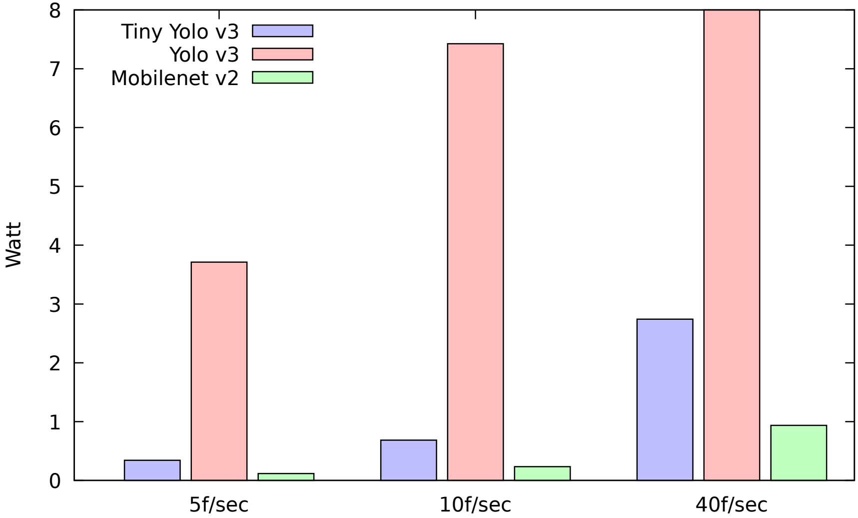
The Mapping Problem

Power Consumption in Inference



VGG16 applied to the ImageNet data set based on published papers.

Power Consumption in Inference



Object detection on the NCS2 platform; own measurements.

What is Special About "Embedded"?

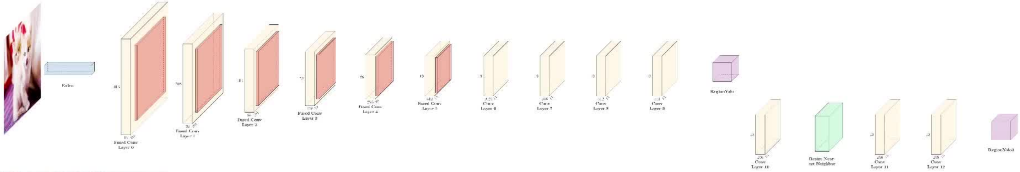
Resource limitations

	Embedded	Server farm
Computation [flop]	$30 - 1800 \cdot 10^{12}$	$86 \cdot 10^{18}$
Memory [bit]	10^{10}	10^{15}
Power [W]	5-100	$10^3 - 10^6$
Energy [Wh]	48-1000	$200 \cdot 10^6$

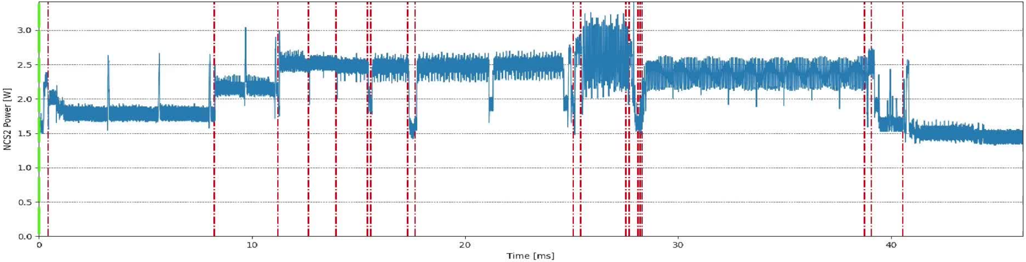
Computation Embedded refers to an Nvidia Jetson Nano running 1 min and 1 hour, respectively.
Computation server refers to the computation needed for the 40 day experiment with AlphaGo Zero
Energy embedded refers to a mobile phone and to a car battery, respectively.
Energy server refers to the 40 day experiment for AlphaGo Zero.

Power and Performance Profiling

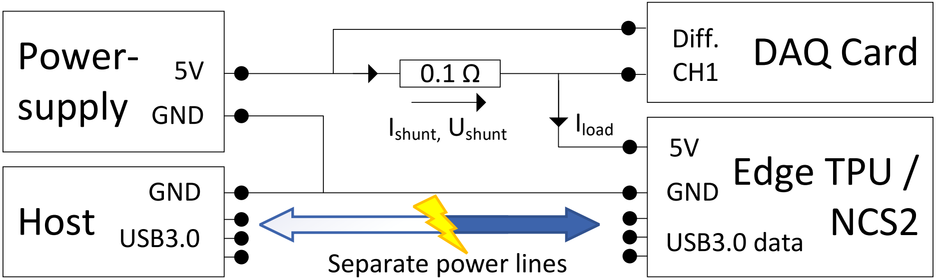
Yolov3-tiny power profile on NCS2

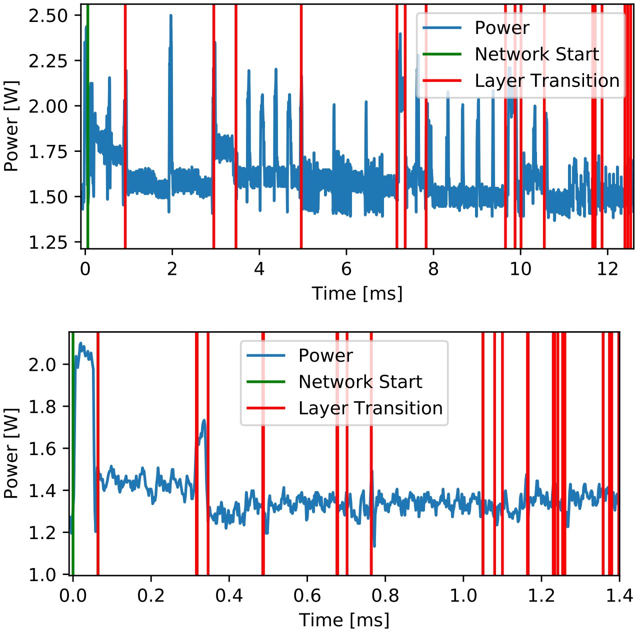


00.000 ms

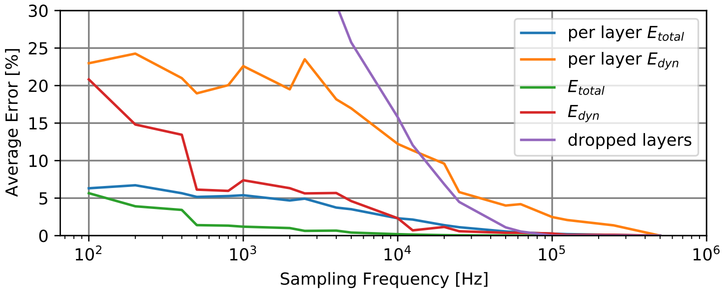


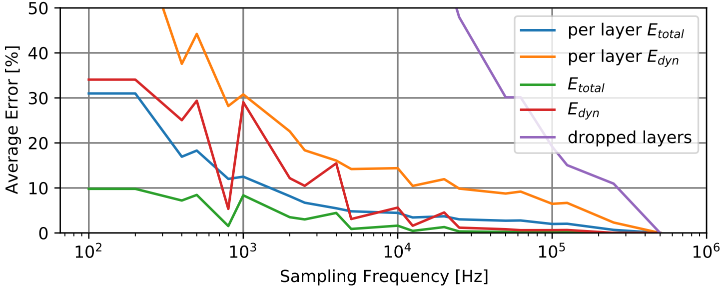
Experimental Setup





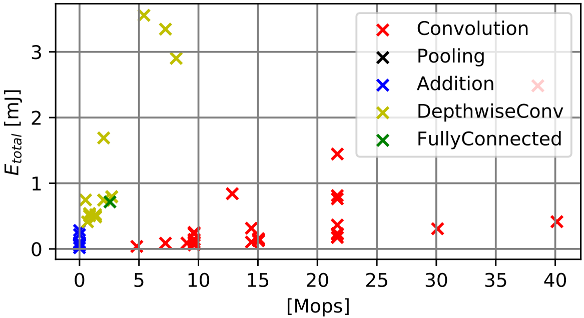
MobileNetV2 on NCS2 and Coral Edge TPU

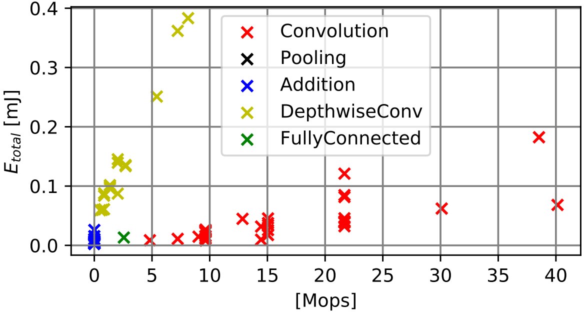




MobileNetV2 on NCS2 and Coral Edge TPU

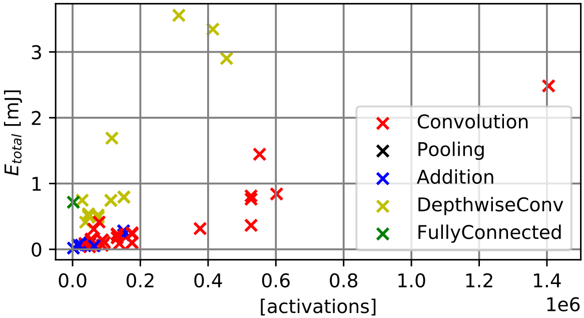
The error in % with respect to 500 kHz sampling frequency.

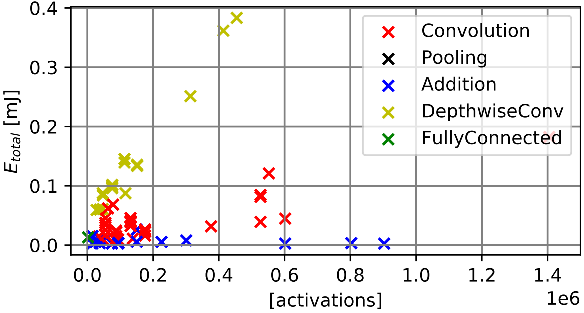




MobileNetV2 on NCS2 and Coral Edge TPU

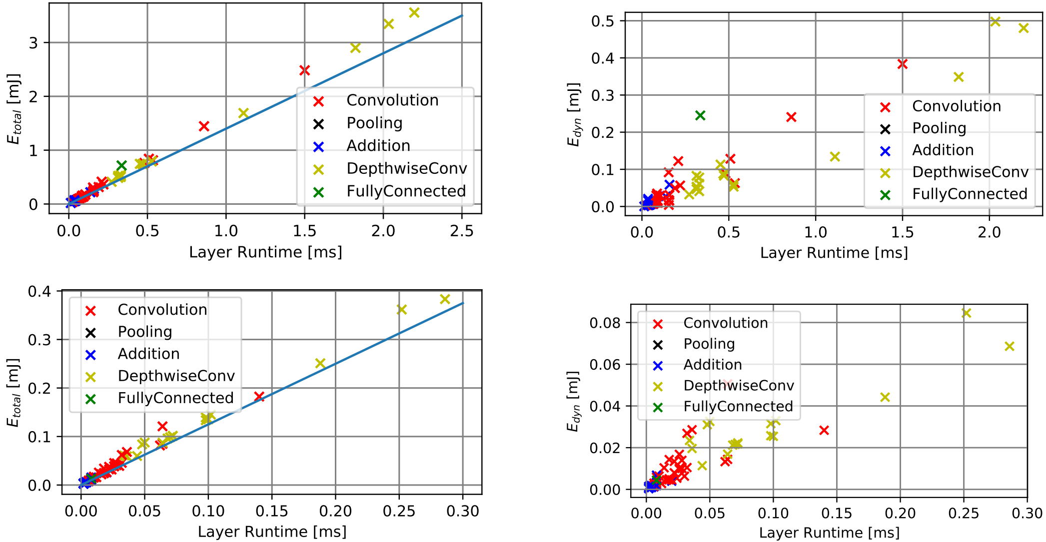
Energy versus number of operations.





MobileNetV2 on NCS2 and Coral Edge TPU

Energy versus number of activations.



MobileNetV2 on NCS2 and Coral Edge TPU; Energy versus latency.

Profiling Results

HW	Network	*nireq*	F_{thr} (fps)	T_{lat} (ms)	P (mW)	E_{total} (mJ)	E_{base} (mJ)	E_{dyn} (mJ)	$E\,/\,Gop$ (mJ)	$E\,/\,Mpar$ (mJ)
NCS2	Tiny YOLOv3 5.6 Gop 8.8 Mpar	1	21.2	41	2165	101.93	65.91	36.02	18.32	11.52
		2	35.3	52	2670	75.55	39.61	35.94	13.58	8.54
		3	43.1	46	2995	69.42	32.45	36.97	12.47	7.85
		4	43.1	44	2954	68.54	32.48	36.06	12.32	7.75
	YOLOv3 65.8 Gop 61.6 Mpar	1	2.6	363	2505	960.92	537.04	423.88	14.69	15.61
		2	4.4	400	3413	769.61	315.69	453.92	11.76	12.50
		3	4.7	425	3615	764.89	296.22	468.67	11.69	12.42
		4	4.9	390	3604	742.50	288.43	454.07	11.35	12.06
	MobileNetV2 0.6 Gop 3.4 Mpar	1	49.3	21	1806	36.60	28.37	8.23	60.84	10.55
		2	87.2	23	2118	24.29	16.06	8.23	40.38	7.00
		3	90.4	31	2164	23.95	15.49	8.46	39.81	6.90
		4	92.4	53	2162	23.39	15.15	8.24	38.88	6.74

HW	Network	*Freq*	F_{thr} (fps)	T_{lat} (ms)	P (mW)	E_{total} (mJ)	E_{base} (mJ)	E_{dyn} (mJ)	$E\,/\,Gop$ (mJ)	$E\,/\,Mpar$ (mJ)
Edge TPU	Tiny YOLOv3	std	46.3	22.3	1407	30.40	22.28	8.12	5.46	3.44
		max	51.0	19.6	1528	29.95	20.21	9.73	5.38	3.39
	YOLOv3	std	6.3	158.3	1519	240.50	163.27	77.23	3.68	3.91
		max	7.0	142.0	1657	235.36	147.29	88.06	3.60	3.82
	MobileNetV2	std	331.3	3.0	1422	4.29	3.11	1.18	7.13	1.24
		max	512.3	1.9	1658	3.23	2.02	1.21	5.37	0.93

Power and Performance Profiling

- NCS2, Edge TPU and Nvidia platforms
- Detailed, per layer latency and power profiling
- Number of operations is a poor predictor for latency and energy
- Latency and energy usage correlate fairly well
- Hardware setting have significant influence
- 100 kHz sampling frequency is required for 5 % accuracy

Matthias Wess, Dominik Dallinger, Daniel Schnöll, Matthias Bittner, Maximilian Götzinger, and Axel Jantsch. "Energy Profiling of DNN Accelerators". In: *Proceedings of the 26th Euromicro Conference on Digital System Design (DSD)*. Durres, Albania, Sept. 2023

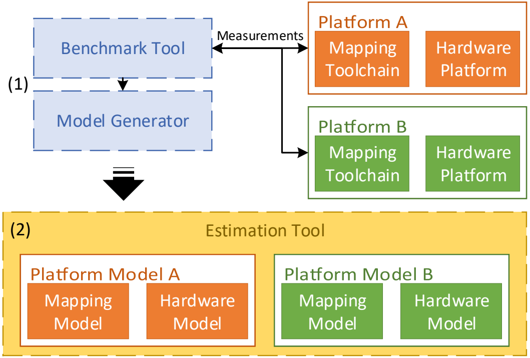
Outline

1. The Mapping Problem
2. Performance Estimation
   - Enhanced Roofline Model
   - Step-wise Linear Model
3. Summary

Performance Estimation

Estimation

- Two leading performance estimation tools: ANNETTE and Blackthorn
- For NCS2, Xilinx FPGA, and Jetson
- Combine analytic, statistical model and partial measurements



Matthias Wess, Marco Ivanov, Christian Unger, Anvesh Nookala, Alexander Wendt, and Axel Jantsch. "ANNETTE: Accurate Neural Network Execution Time Estimation With Stacked Models". In: *IEEE Access* 9 (2021), pages 3545–3556

Martin Lechner and Axel Jantsch. "Blackthorn: Latency Estimation Framework for CNNs on Embedded Nvidia Platforms". In: *IEEE Access* (2021)

Enhanced Roofline Model

Roofline Model

- Performance bound model for multi-core processors

Samuel Williams, Andrew Waterman, and David Patterson. "Roofline: an insightful visual performance model for multicore architectures". In: *Commun. ACM* 52.4 (Apr. 2009), pages 65–76

Roofline Model

- Performance bound model for multi-core processors
- Bound and bottleneck analysis

Samuel Williams, Andrew Waterman, and David Patterson. "Roofline: an insightful visual performance model for multicore architectures". In: *Commun. ACM* 52.4 (Apr. 2009), pages 65–76

Roofline Model

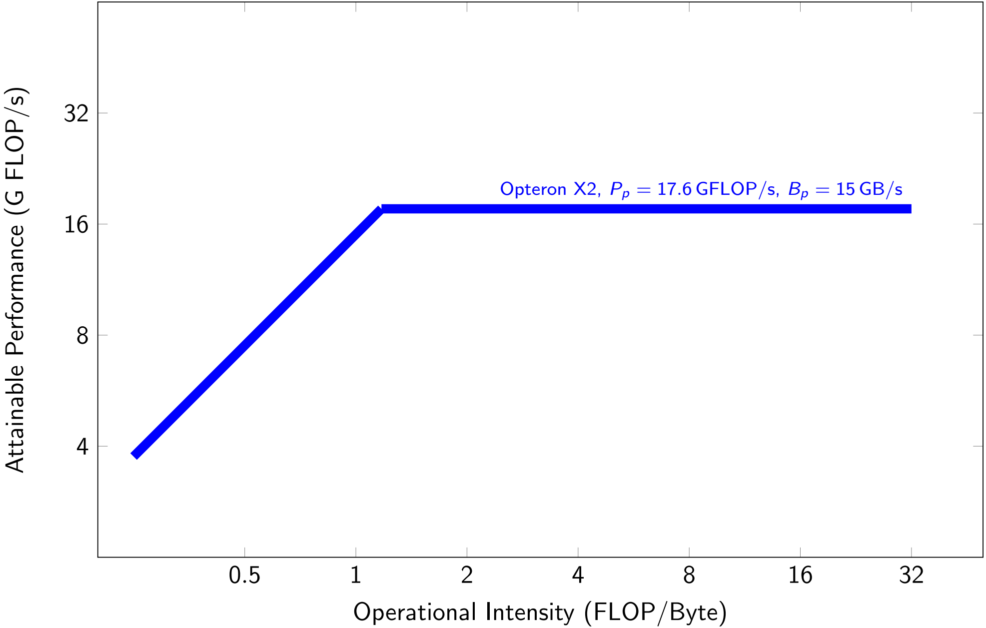
- Performance bound model for multi-core processors
- Bound and bottleneck analysis
- Main bounds due to
  - P_p: Maximum attainable operations per second (in FLOP/second)
  - B_p: Maximum attainable memory traffic between processor (including cache hierarchy) and DRAM (in Byte/second)

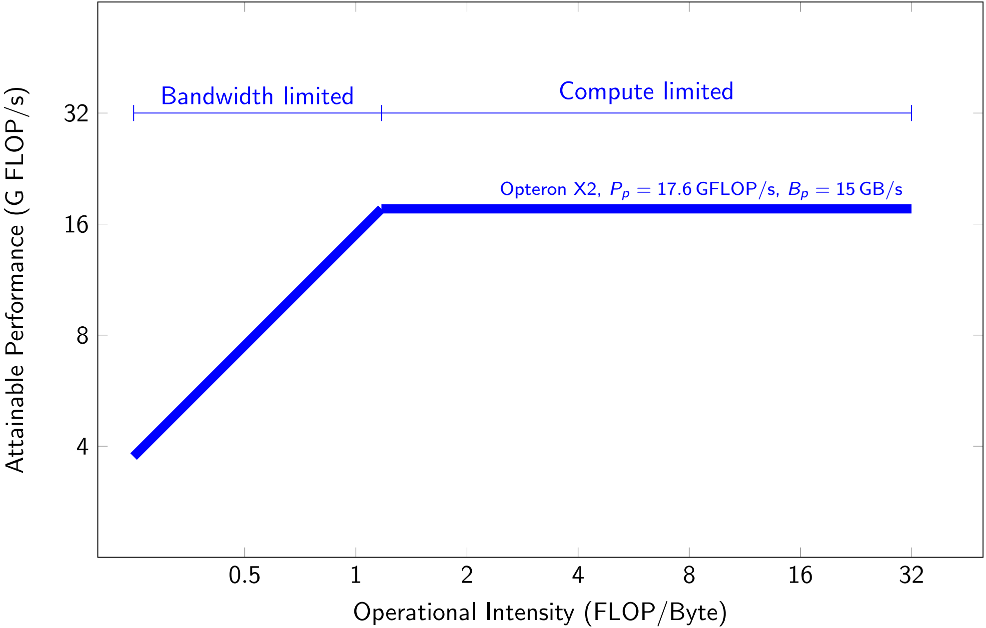
Samuel Williams, Andrew Waterman, and David Patterson. "Roofline: an insightful visual performance model for multicore architectures". In: *Commun. ACM* 52.4 (Apr. 2009), pages 65–76

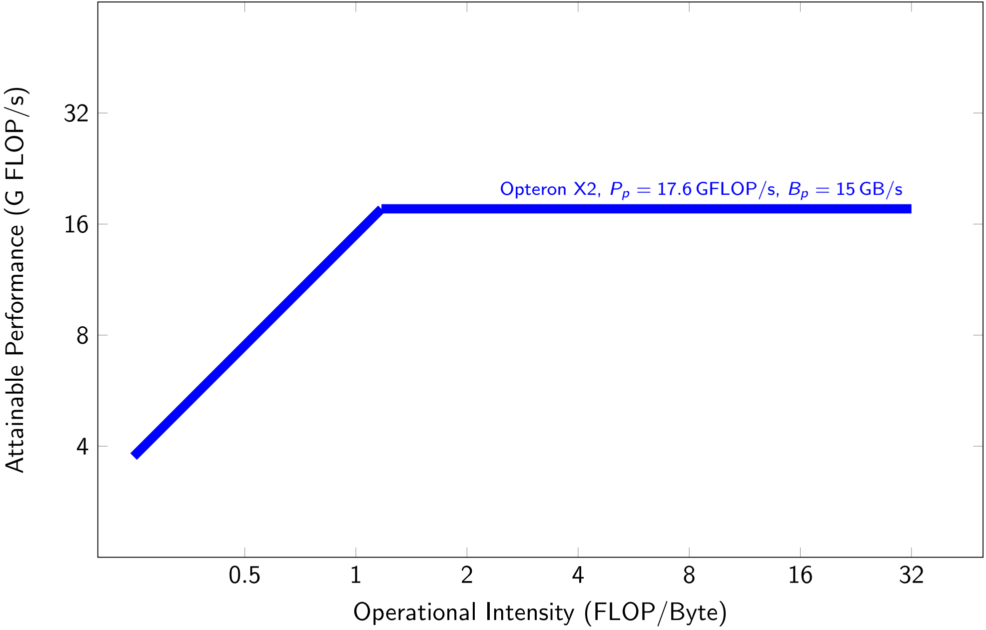
Roofline Model

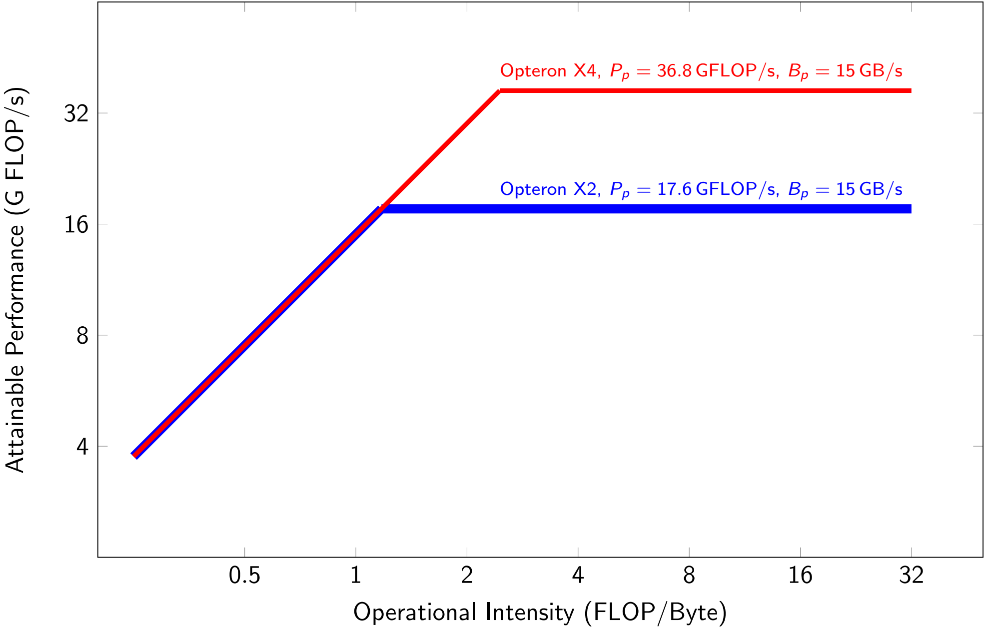
- Performance bound model for multi-core processors
- Bound and bottleneck analysis
- Main bounds due to
  - P_p: Maximum attainable operations per second (in FLOP/second)
  - B_p: Maximum attainable memory traffic between processor (including cache hierarchy) and DRAM (in Byte/second)
- $o = \frac{P}{B}$: Operational intensity is performance per memory traffic (in FLOP/Byte)

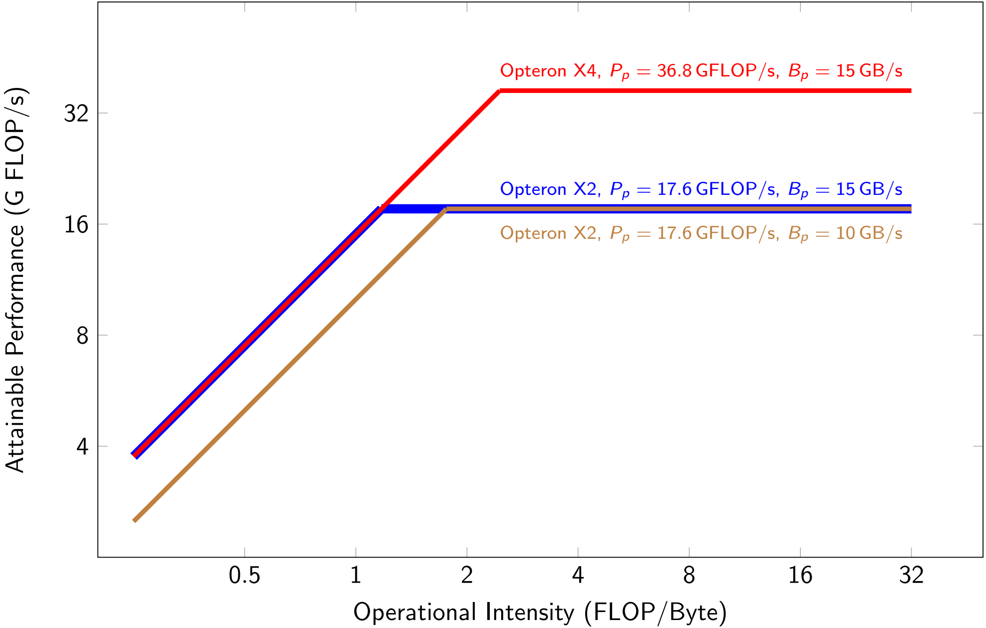
Samuel Williams, Andrew Waterman, and David Patterson. "Roofline: an insightful visual performance model for multicore architectures". In: *Commun. ACM* 52.4 (Apr. 2009), pages 65–76


Opteron X2, $P_p = 17.6\,\text{GFLOP/s}$, $B_p = 15\,\text{GB/s}$
Attainable Performance (G FLOP/s)
32
16
8
4
0.5
1
2
4
8
16
32
Operational Intensity (FLOP/Byte)


Bandwidth limited
Compute limited
Opteron X2, $P_p = 17.6\,\text{GFLOP/s}$, $B_p = 15\,\text{GB/s}$
32
16
8
4
0.5
1
2
4
8
16
32
Attainable Performance (G FLOP/s)
Operational Intensity (FLOP/Byte)


Opteron X2, $P_p = 17.6\,\mathrm{GFLOP/s}$, $B_p = 15\,\mathrm{GB/s}$
Attainable Performance (G FLOP/s)
32
16
8
4
0.5
1
2
4
8
16
32
Operational Intensity (FLOP/Byte)


Opteron X4, $P_p = 36.8\,\text{GFLOP/s}$, $B_p = 15\,\text{GB/s}$
Opteron X2, $P_p = 17.6\,\text{GFLOP/s}$, $B_p = 15\,\text{GB/s}$
Attainable Performance (G FLOP/s)
32
16
8
4
0.5
1
2
4
8
16
32
Operational Intensity (FLOP/Byte)


Opteron X4, $P_p = 36.8\,\text{GFLOP/s}$, $B_p = 15\,\text{GB/s}$
Opteron X2, $P_p = 17.6\,\text{GFLOP/s}$, $B_p = 15\,\text{GB/s}$
Opteron X2, $P_p = 17.6\,\text{GFLOP/s}$, $B_p = 10\,\text{GB/s}$
Attainable Performance (G FLOP/s)
32
16
8
4
0.5
1
2
4
8
16
32
Operational Intensity (FLOP/Byte)

Refined Roofline Model

$$\hat{T}_{\text{roof}_n}(f_n, D_n) = \max\left(\frac{f_n}{P_{\text{peak}}}, \frac{D_n}{B_{\text{peak}}}\right)$$

with

n	… Layer
$\hat{T}_{\text{roof}_n}$	… Latency for layer n
f_n	… No of operations
D_n	… No of bytes to be transferred
P_{peak}	… Peak performance (FLOP/s))
B_{peak}	… Peak bandwidth (Byte/s)

Matthias Wess, Marco Ivanov, Christian Unger, Anvesh Nookala, Alexander Wendt, and Axel Jantsch. "ANNETTE: Accurate Neural Network Execution Time Estimation With Stacked Models". In: *IEEE Access* 9 (2021), pages 3545–3556

Refined Roofline Model

$$\hat{T}_{\mathrm{ref}_n}(f_n, D_n) = max\left(\frac{f_n}{P_{\mathrm{peak}}\, u_{\mathrm{eff}_n}}, \frac{D_n}{B_{\mathrm{peak}}}\right)$$

$$u_{\mathrm{eff}}(\vec{x}) = \prod_{i=1}^{A} \frac{x_i/s_i}{\lceil x_i/s_i \rceil}$$

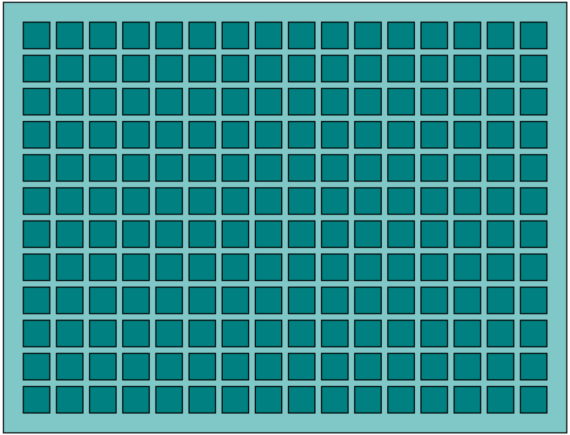
with

u_{eff_n} ... utilization efficiency
$\vec{s}$ ... Number of resources
$\vec{x}$ ... Number of operations

Refined Roofline Model

12x6x128 Input Feature Map
1x1 Convolution
256 Output Feature Map

16x12 PE Array



$$
\begin{aligned}
\vec{s} &= (16 \times 12) \\
\vec{x} &= (12 \times 6 \times 128 \times 256 \times 1 \times 1) \\
u_{\text{eff}}(\vec{x}) &= \prod_{i=1}^{A} \frac{x_i/s_i}{\lceil x_i/s_i \rceil} = \frac{12/12}{1} \cdot \frac{6/16}{1} = 0.375
\end{aligned}
$$

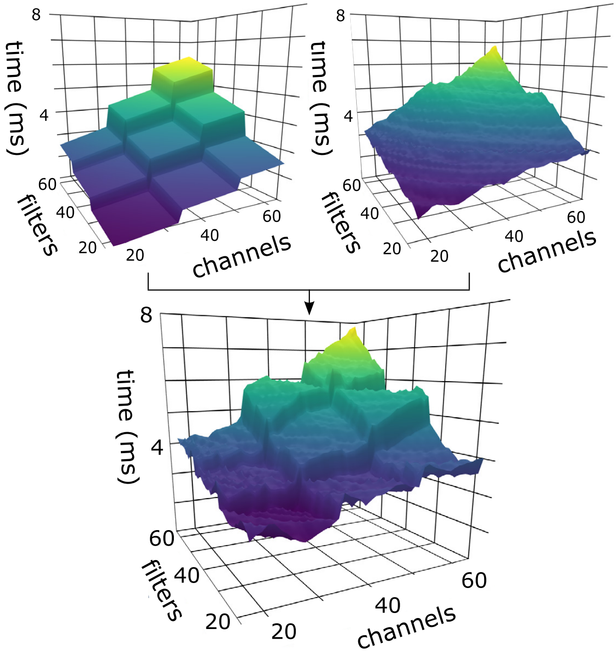
Statistical Model

- Regresssion model to estimate utilization efficiency u_{stat}
- Feature vector for 2D convolution:
  $(h, w, c, f, k_h, k_w, \text{stride}, \#\text{ops}, \#\text{in}, \#\text{out}, \#\text{weights})$.
- Random forest prediction method

$$\hat{T}_{\text{stat}_n}(f_n, D_n) = max\left(\frac{f_n}{P_{\text{peak}}u_{\text{stat}_n}}, \frac{D_n}{B_{\text{peak}}}\right)$$

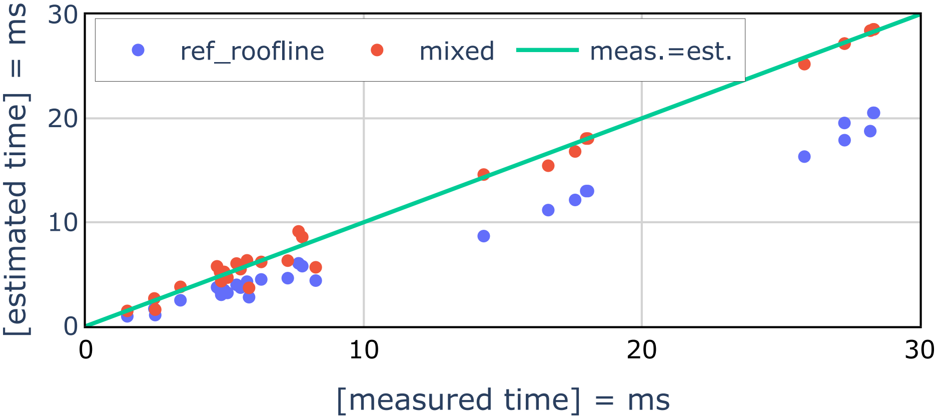
Mixed Model

$$\hat{T}_{\text{mixed}_n}(f_n, D_n) = max\left(\frac{f_n}{P_{\text{peak}}\,u_{\text{eff}_n}\,u_{\text{stat}_n}}, \frac{D_n}{B_{\text{peak}}}\right)$$



8
4
time (ms)
60
40
20
filters
20
40
60
channels
8
4
time (ms)
60
40
20
filters
20
40
60
channels
8
4
time (ms)
60
40
20
filters
20
40
60
channels

Mixed Model



Test subset of NASBench data set
NCS2 platform

Mixed Roofline Model - Results

Device	Model Type	Measured (ms)	MAE (ms)	MAPE (%)
NCS2	Roofline	226.3	67.8	30.0
	Ref. Roofline	219.7	64.9	29.6
	Statistical	233.8	18.5	7.9
	Mixed	200.9	**15.0**	**7.4**
ZCU102	Roofline	19.8	6.1	30.9
	Ref. Roofline	15.0	4.1	27.2
	Statistical	41.7	2.5	6.0
	Mixed	25.6	**0.9**	**3.5**

Network execution time for 12 networks

(4 Inception, 2 ResNet, 1 FPN, 1 Open Pose, 2 MobileNet, 2 Yolo)

MAE ... Mean Absolute Error

MAPE ... Mean Absolute Percentage Error

Matthias Wess, Marco Ivanov, Christian Unger, Anvesh Nookala, Alexander Wendt, and Axel Jantsch. "ANNETTE: Accurate Neural Network Execution Time Estimation With Stacked Models". In: *IEEE Access* 9 (2021), pages 3545–3556

Step-wise Linear Model

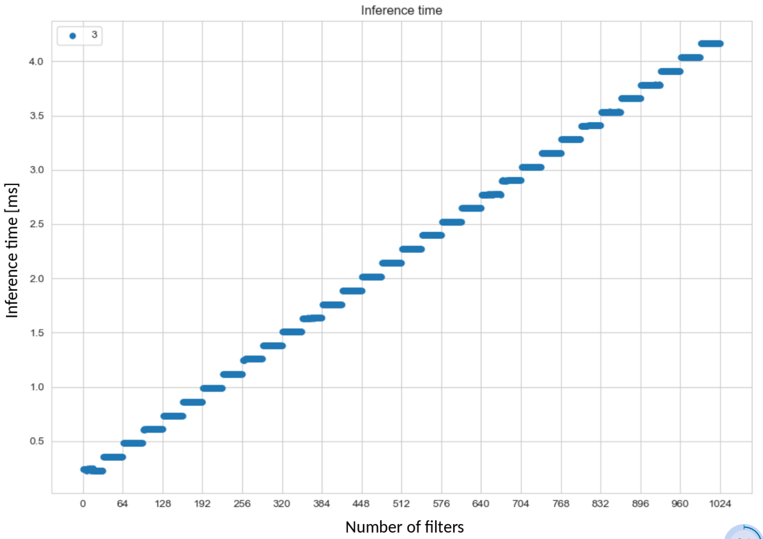
Inference Run Time Estimation

Assumption:

- Inference time as a function of problem size is a combination of step and linear functions due to limited parallel resources.
-

Example:

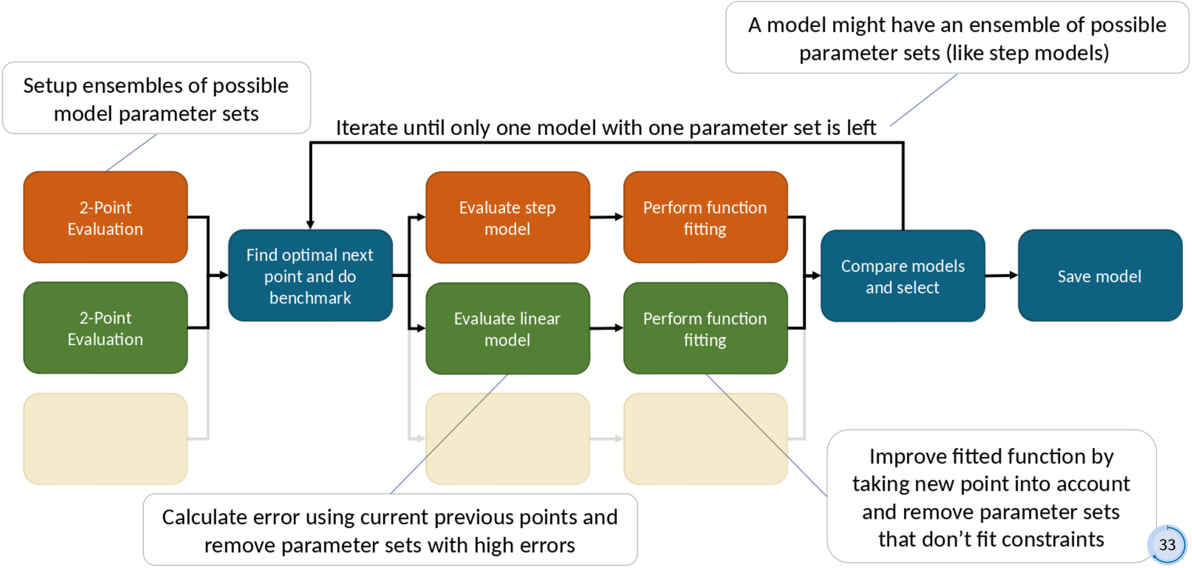
- Single convolutional layer sweep
- 32x32x64 with k filter and kernel size 3



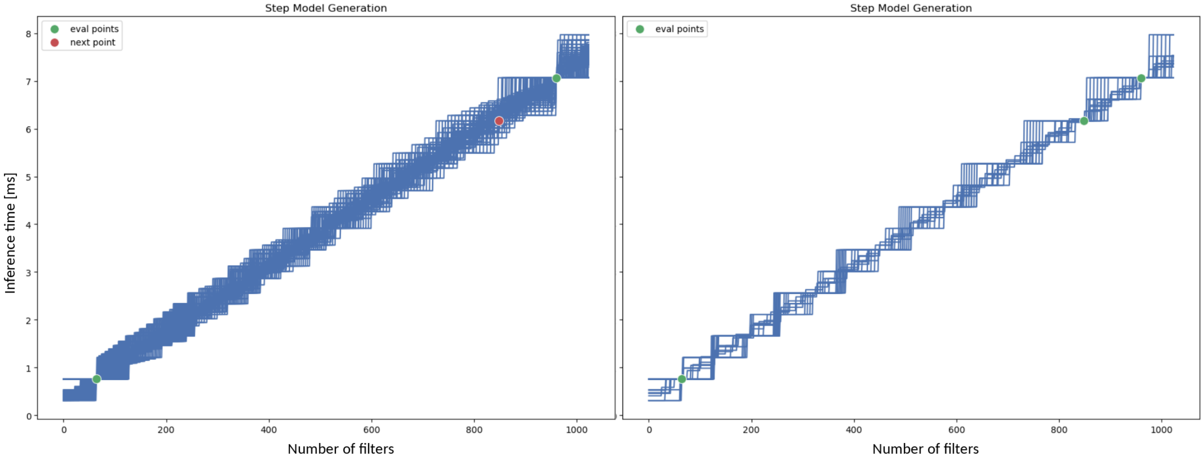
Inference Run Time Estimation

- Assumption:
  The inference time can be approximated by a combination of linear and step functions for each dimension, such as filter, channels, etc.
- Determining the function based on selected measurements
- Goals: automatic computation of estimation functions for latency, power consumption and various platforms.

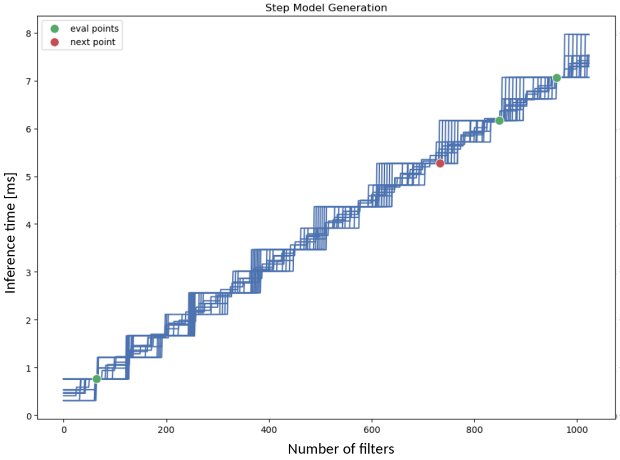
Automatic Estimation Function Generation

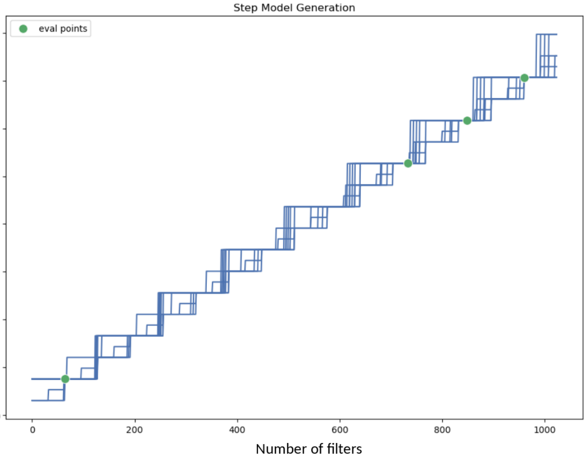


Iterative Refinement

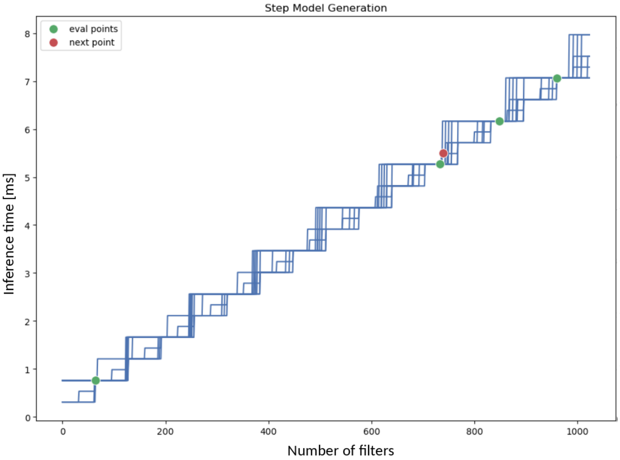


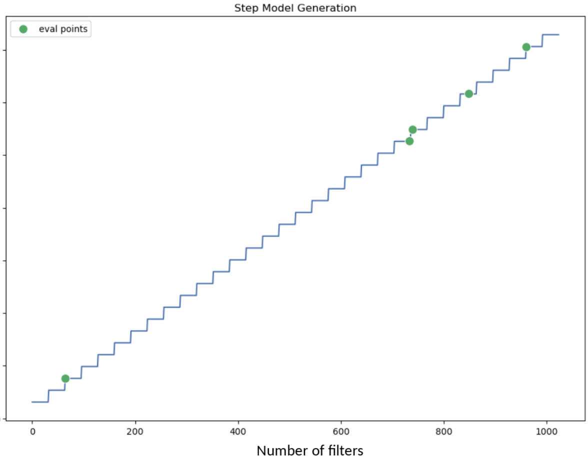
Iterative Refinement





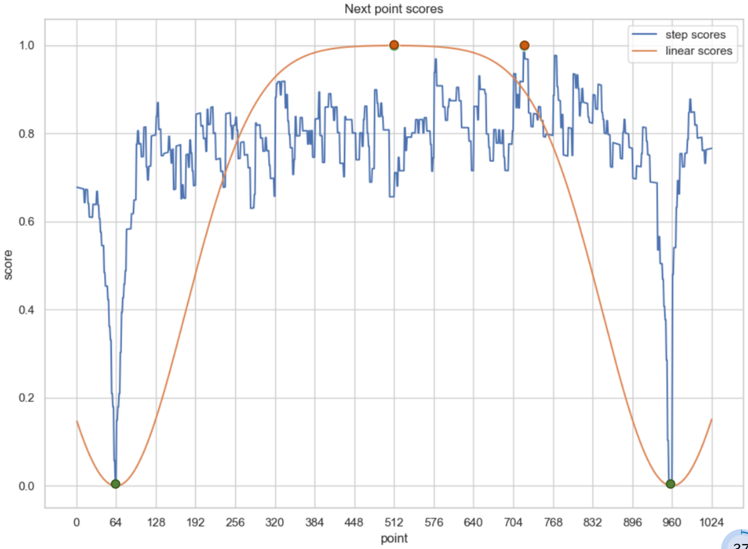
Iterative Refinement



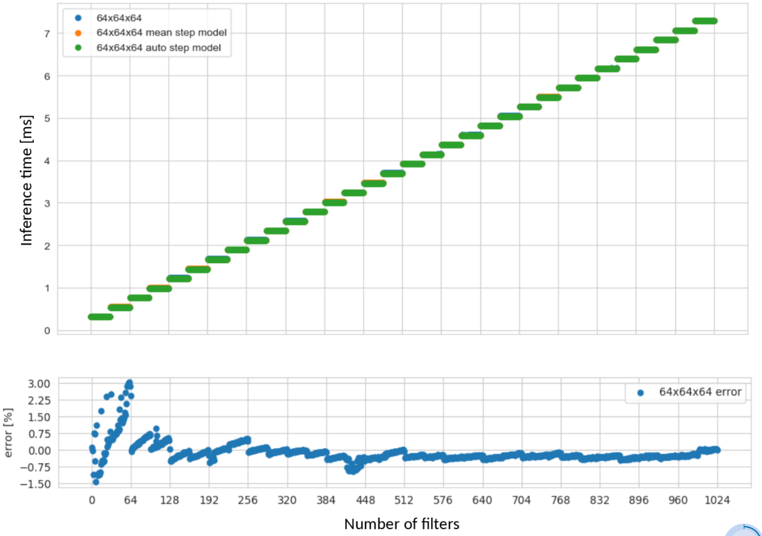


Next Point Selection

- Linear function criteria:
  - Point furthest away from previous points
- Step function criteria
  - Point with most unique discrete levels
  - Point with largest range of values
  - Point furthest away from previous points
- Next point selection: Point with highest score

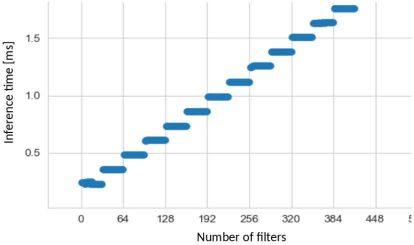


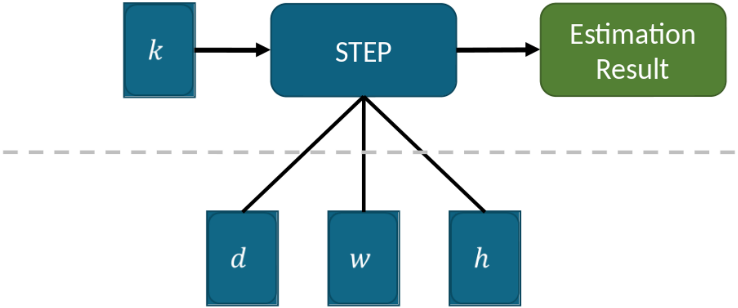
- Results after 3 iterations (5 meausurment points)
- Execution times:
  - Full sweep: 3-4 h
  - Proposed approach: 2-5 minutes



2D Example

- Phase 1: Estimate function in single dimension: number of filters
- Result: step function

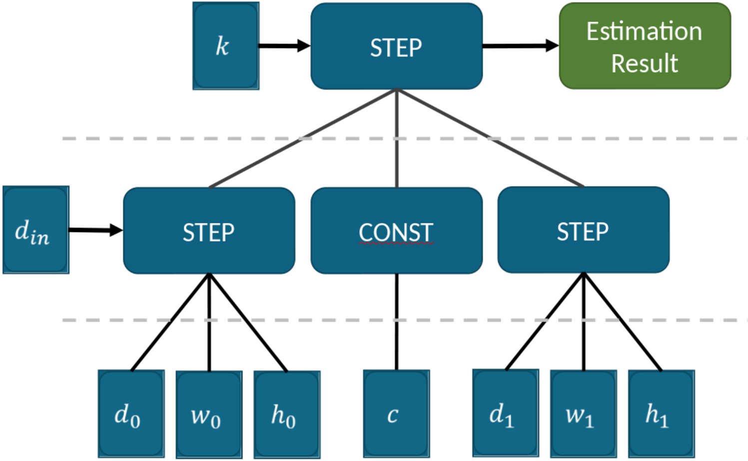




$$0{,}216 + \left\lfloor \frac{k-1}{32} \right\rfloor 0{,}01286$$

2D Example

- Phase 2: Test how d, w and h behave in the next dimension
- Next dimension: input channels d_{in}
- Result:
  - Step function: $d_{0=0.1418}$, $w_0 = 8$, $\mathrm{h}_0 = 0.0106$
  - Constant: $c = 32$
  - Step function: $d_1 = 0.044$, $w_1 = 8$, $h_1 = 0.0121$



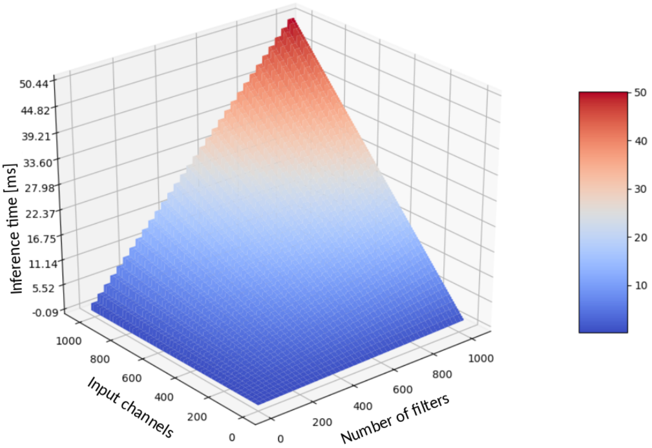
$$0.1418 + \left\lceil \frac{d_{in} - 1}{8} \right\rceil 0.0106 + \left\lceil \frac{k - 1}{32} \right\rceil \left(0.044 + \left\lceil \frac{d_{in} - 1}{8} \right\rceil 0.0121 \right)$$

2D Example

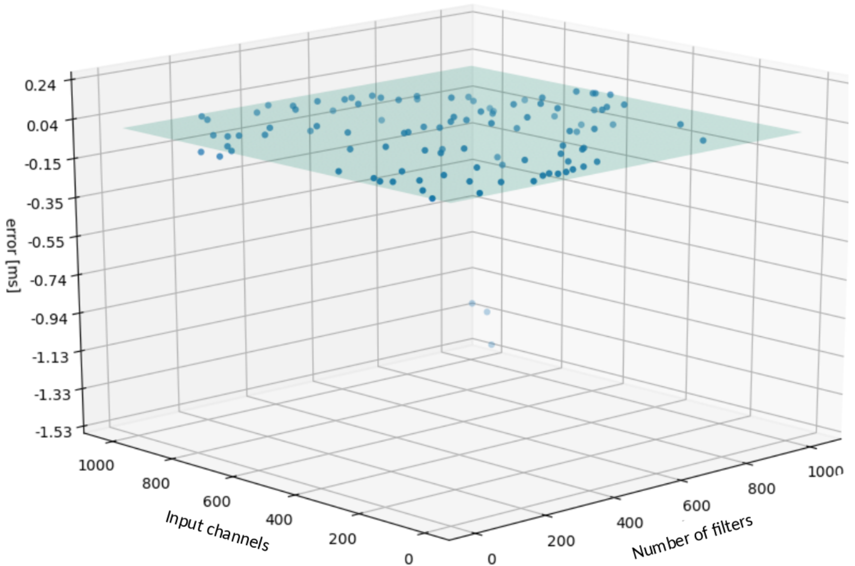
Generated model:

$$
\begin{aligned}
&f(d_{\text{in}}, k) \\
&\quad = 0.1418 + \lfloor \frac{d_{\text{in}} - 1}{8} \rfloor 0.0106 \\
&\quad + \lfloor \frac{k-1}{32} \left(0.044 + \lfloor \frac{d_{\text{in}} - 1}{8} \rfloor 0.0121 \right)
\end{aligned}
$$

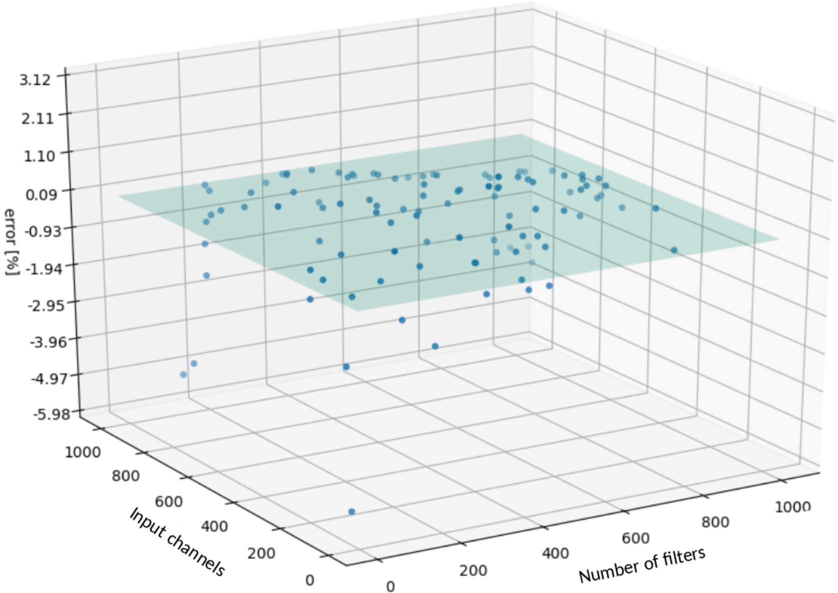
- Meausrement points: 112
- Execution time: 32 minutes



2D Example - Error



2D Example - Error

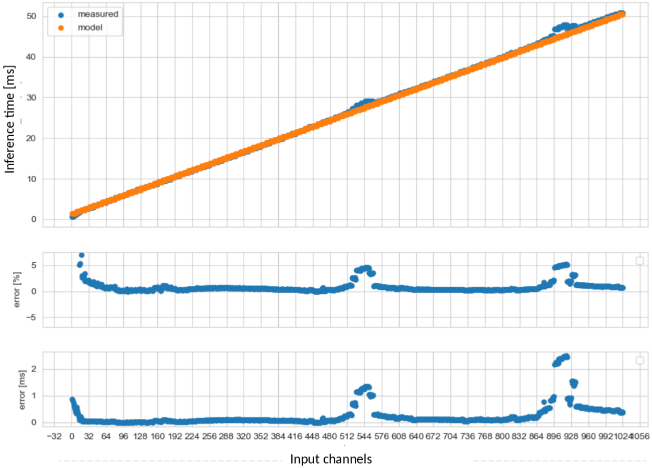


2D Example - Error

Slice through 2D plane at $k = 1024$

$$f(d_{\text{in}}, k)$$
$$= 0.1418 + \lfloor \frac{d_{\text{in}} - 1}{8} \rfloor 0.0106$$
$$+ \lfloor \frac{k-1}{32} \left(0.044 + \lfloor \frac{d_{\text{in}} - 1}{8} \rfloor 0.0121\right)$$

$$f(d_{\text{in}}, 1024)$$
$$= 1.5058 + \lfloor \frac{d_{\text{in}} - 1}{8} \rfloor 0.3857$$

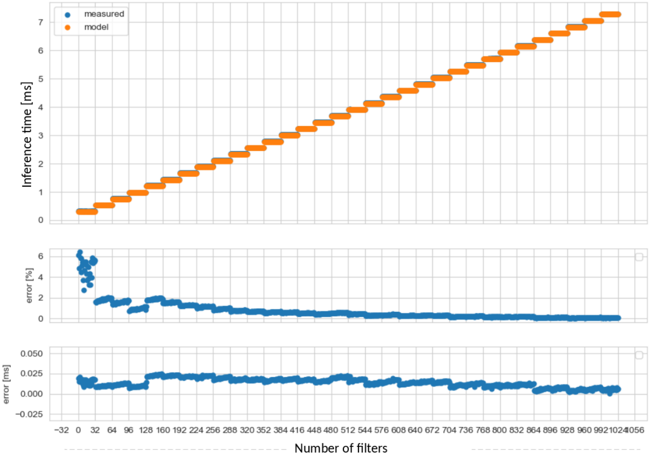


2D Example - Error

Slice through 2D plane at $d_{in} = 128$

$$f(d_{in}, k)$$
$$= 0.1418 + \lfloor \frac{d_{in} - 1}{8} \rfloor 0.0106$$
$$+ \lfloor \frac{k-1}{32} \rfloor \left(0.044 + \lfloor \frac{d_{in} - 1}{8} \rfloor 0.0121\right)$$

$$f(128, k)$$
$$= 0.3008 + \lfloor \frac{k-1}{32} \rfloor 0.2255$$



Blackthorn Estimation Results

Device	Network	Measured (ms)	MAE (ms)	MAPE (%)
Jetson Nano	AlexNet	27.8	1.5	5.5
	VGG16	154.9	0.7	0.5
	ResNet 50	49.2	1.1	2.3
	MobileNetV2	13.7	0.5	3.6
Jetson TX2	AlexNet	11.2	0.8	6.7
	VGG16	61.2	0.9	1.4
	ResNet 50	21.4	1.0	4.8
	MobileNetV2	6.7	0.3	4.2

Network execution time
MAE ... Mean Absolute Error
MAPE ... Mean Absolute Percentage Error

Martin Lechner and Axel Jantsch. "Blackthorn: Latency Estimation Framework for CNNs on Embedded Nvidia Platforms". In: *IEEE Access* (2021)

Outline

SUMMARY

Latency Estimation Summary

- Exploiting the discrete nature of HW resources
- Systematic benchmarking of a platform and a set of network layer types
- Fast estimation function for latency for any new network with known layer types
- Results for several platforms are robust

Network	Estimation Error [%]			
	NCS2	ZCU102	Jetson Nano	Jetson TX2
YoloV3	4.1	3.2	-	-
MobileNetV2	4.3	4.2	3.6	4.2
ResNet50	8.2	1.2	2.4	4.8
FPN Net	9.3	7.5	-	-
AlexNet	5.2	4.8	5.5	6.6
VGG16	11.3	6.2	0.5	1.4

https://eml.ict.tuwien.ac.at/

Christian Doppler Laboratory

Embedded Machine Learning

References

[1] Matthias Wess, Dominik Dallinger, Daniel Schnöll, Matthias Bittner, Maximilian Götzinger, and Axel Jantsch. "Energy Profiling of DNN Accelerators". In: *Proceedings of the 26th Euromicro Conference on Digital System Design (DSD)*. Durres, Albania, Sept. 2023.

[2] Matthias Wess, Marco Ivanov, Christian Unger, Anvesh Nookala, Alexander Wendt, and Axel Jantsch. "ANNETTE: Accurate Neural Network Execution Time Estimation With Stacked Models". In: *IEEE Access* 9 (2021), pages 3545–3556.

[3] Martin Lechner and Axel Jantsch. "Blackthorn: Latency Estimation Framework for CNNs on Embedded Nvidia Platforms". In: *IEEE Access* (2021).

[4] Samuel Williams, Andrew Waterman, and David Patterson. "Roofline: an insightful visual performance model for multicore architectures". In: *Commun. ACM* 52.4 (Apr. 2009), pages 65–76.


TECHNISCHE HOCHSCHULE