Explainable-DSE: Agile & Explainable Exploration of Efficient HW/SW Codesigns of DL Accelerators Semiconductor

Shail Dave¹, Tony Nowatzki², Aviral Shrivastava¹

¹School of Computing and AI, Arizona State University; ²School of Computer Science, University of California, Los Angeles

Motivation

Develop framework for explainable DSE of deep learning accelerators that reasons about underlying inefficiencies in designs, achieve efficient designs, takes short time, and can work with several domains.

Background

- Effective design space exploration (DSE) requires achieving efficient solutions satisfying constraints under practical exploration budgets
- Deep learning accelerator design space can be vast
- 10^14 hardware parameter solutions
- 10^12—10^36 mappings of DNNs layers on each AIHW architecture
- Recent industrial/academic approaches use black-box DSEs

Research

Corporation

- Evolutionary algorithms
- ML-based approaches

Table 1: Design Space for Edge DNN Accelerators.

Data: int16; Freq. 500 MHz; Constraints: Throughput>=40/10 FPS (vision light/heavy), 120/530/176k samples/second (NLP: Transformer/BERT/wav2vec2); Area < 75 mm²; Max. power < 4W.

Objective:	Minimize	latency.

Parameter	Values	Options
PEs	64, 128,, 4096	7
L1 buffer (B)	8, 16,, 1024	8
L2 buffer (kB)	64, 128,, 4096	7
Offchip bandwidth	1024, 2048, 4096, 6400, 8192,	10
(MBPS)	12800, 19200, 25600, 38400, 51200	10
NOC datawidth	16*i; i: [1, 16]	16
Physical unicast (\times 4)	PEs*i / 64; i: [1, 64]	64 ⁴
Virtual unicast (×4)	2 ³ⁱ ; i: [0, 3]	4 ⁴



- Process single cost value for a DNN vs. per-layer costs and do not leverage information about domain-specific bottlenecks
- They require excessive trials (thousands), which leads to
 - Low Efficiency of obtained solutions (several-fold)
 - Low Feasibility (most of acquired solutions do not meet constraints) Cannot do Runtime/Practical DSE (Takes days—weeks)

Cost models take millisecondsminutes; Practical explorations can afford only 1000s of iterations

Explainability can be enabled by domain-awareness and bottleneck analysis Prior works lack formal specification of how to express bottleneck analysis in

a generic manner for an effectual design space exploration

DSE Using Bottleneck Analysis & API for Expressing Domain-Specific Bottleneck Models

128,

64,

51.2,

48,

500>

(d) Acquiring Bottleneck-Mitigating Candidates



•Through API, architects/tools can Specify Bottleneck Graph of target costs and appropriate Scaling of Parameters



 Workflow example for Solution Acquisitions addressing Multiple **Bottlenecks and Constraints-Budget-Awareness**

A DNN-18. Unique	Layer ID	% total latency	Bottlenec	k	Related De Paramete	sign ers	Value	
	1	16.87%	NoC time	e L1_s lir	ize, NoC1_ı ıks, NoCs_ı	unicast_ width	256, 2, 93	
	2	11.02%	NoC time	e L1_s lir	ize, NoC1_u nks, NoCs_v	unicast_ width	256, 2, 66	
for distribution	3	11.55%	DMA time	e	L2_size		128	
E Controller reduction NoCs	4 🚺	11.09%	DMA time	e	L2_size		128	
NPU L2 Buffer	5 🤇	13.16%	DMA time	e	L2_size		128	
{PEs: 256. L1 size: 128B. L2 size: 64KB.	6	06.78%	Compute ti	me	PEs		343	
offchip_BW: 51.2GBPS, NoC1_unicast_links: 1,	7	10.32%	NoC time	L1_s	ize, NoC1_u nks, NoCs_v	unicast_ width	256, 3, 74	
NoCz_unicust_iniks. 1, NoCs_unicust_iniks. 04,	8	09.66%	DMA time	e	L2_size		128	
Nocs_width: 480, frequency. 500 Winz	9	09.55%	DMA time	е	L2_size		128	
(a) Example of a workload and an DNN accelerator	(b) Analyz	ing Bottle	necks in	Workloa	d Execu	tions	
Current Candidate Solutions Acquired Solution (S) for Evaluations (CS) <256, Best		ed Scenario 1: Constraints Unmet Check Utilized Budgets of Constraints Budget: Cost X's Value ÷ Constraint X's Value						
128 , → <256, 256 , 64, 51.2, 1, 1, 64, 48, 500> Solution		A Constra	aint is Met V	Vhen Buc	lget <= 1	X: Do	on't Ca	
64, → <256, 128, 128, 51.2, 1, 1, 64, 48, 500> Is Updated After	Candi #	date Ob	oj. Constr1	Constr2	Constr3	Average	# Me	
1, → <256, 128, 64, 51.2, 4 , 1, 64, 48, 500> Evaluating	1	. X	1.719	0.104	0.196	0.673	2/3	
1, 64, Values Rounded Up to Design Space Range Acquired	2	x	1.592	0.097	0.171	0.620	2/3	
48, → <256, 128, 64, 51.2, 1, 1, 64, 80, 500> Candidates	(3) x	1.485	0.081	0.168	0.578	2/3	
500> One acquisition attempt.	4	. Х	1.889	0.091	0.172	0.717	2/3	

	Consider layers tha the total o	estimations t contribute cost. Set K=5	for bott to at la 5, thresh	lenecks of st %three to be a second strain of the	of Top-K shold of $\frac{1}{L} \times 100\%$	ý.
	L1_siz	e L2_size	e NC un lin	DC1_ icast_ ks	NOCs_ width	
	$\min\binom{25}{25}$	$\binom{6}{6}{6} \min \begin{pmatrix} 12\\12\\12\\12 \end{pmatrix}$	8, 8, 8) mi	$n\binom{2}{2}{2}$	$\min\binom{93,}{66}$	
	= 256	5 = 128	3 =	= 2	= 66	
	(c) Aggre in Mult	egating Pre i-Functiona	dictior al Work	is for Bo cload Exc	ttleneck ecutions	S
	(c) Aggre in Mult or	egating Pre i-Functiona Scenario 2: Check for at Low	dictior al Work All Co Low Va Constr	is for Bo doad Ex nstraint lue of Ol aints-Bu	ottleneck ecutions is are Me bjective dget	s t
re t	(c) Aggre in Mult or	egating Pre i-Functiona Scenario 2: Check for at Low Candidate	dictior al Work All Co Low Va Constr Obj.	ns for Bo cload Exe nstraint lue of Ol aints-Bu Constr. Budget	ottlenecks ecutions is are Me bjective dget Obj. x Constr. Budget	s t
re t	(c) Aggre in Mult or	egating Pre i-Functiona Scenario 2: Check for at Low Candidate #	dictior al Work All Co Low Va Constr Obj. 15.7	ns for Bo doad Exe nstraint lue of Ol aints-Bu Constr. Budget 0.58	ottlenecks ecutions is are Me bjective dget Obj. x Constr. Budget 9.20	s

15.8

38.3

0.42

0.42

6.63

16.23

(e) Constraints-Aware Update of Best Solution

- Bottleneck Path (Bytes) (Bytes/Cycle)

return newConfig

Results



Consistent/Quick Objective Reduction



- Constraints-budget-aware DSE majorly explores feasible solutions (87% vs. 21%) for non-explainable DSEs)
- Including software design space in the exploration enables 4.24x better solutions and tightly coupled codesign (search time increases from 21 to 64 minutes; from 16

Conclusions

Know what designs you explore and why!

- Black-box DSE cannot explain why solutions incur high costs and underlying inefficiencies or goodness of potential configurations to try.
- Explainability can be enabled with bottleneck analysis of costs and gray-box estimations for mitigations.
- Formalizing specification for bottleneck models can help applying DSE to broader workload domains.
- Address diverse bottlenecks in multi-workload executions by aggregating mitigation predictions.
- With Explainability, DSE can find several-fold efficient solutions at runtime!

Future Directions

- Automate bottleneck model generation for a variety of domain-specific architectures
- Introducing bottleneck mitigations in ML-based DSE
- Overcome local-optima-convergence due to
- greediness for mitigating primary bottlenecks

Publications

Explainable-DSE: Agile & Explainable HW/SW Codesign Exploration using Bottleneck Analysis. In ACM International



Website and Related Material:



