ARTIFICIAL NEURAL NETWORK AND ACCELERATOR CO-DESIGN USING EVOLUTIONARY ALGORITHMS

PHILIP COLANGELO[1,3], OREN SEGAL[2], ALEX SPEICHER[2], MARTIN MARGALA[1]

[1] UNIVERSITY OF MASSACHUSETTS LOWELL
[2] HOFSTRA UNIVERSITY
[3] INTEL

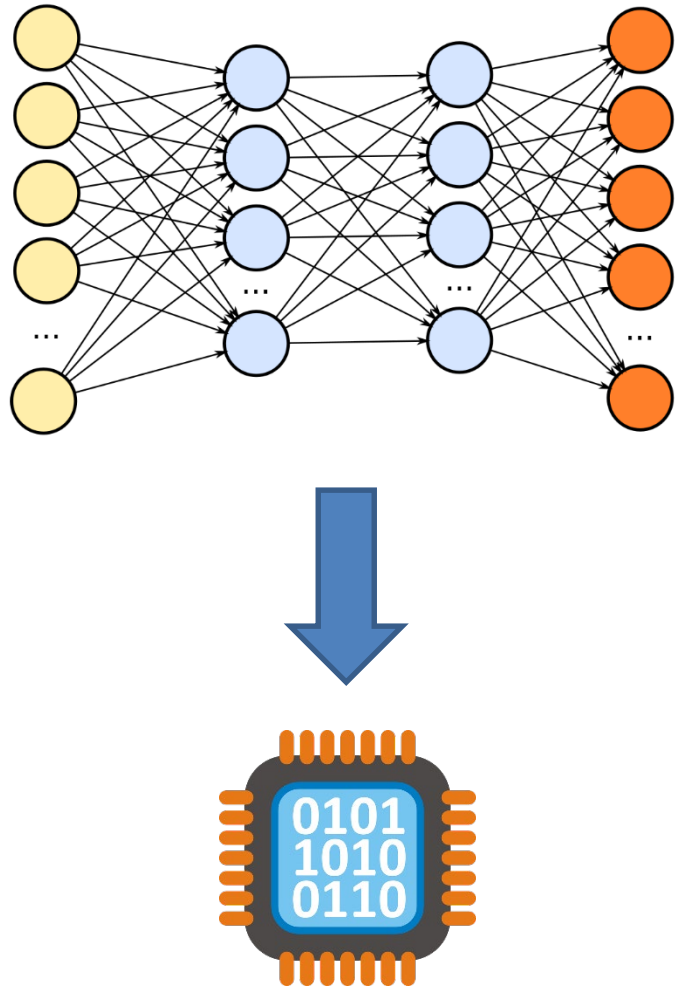
LEGAL DISCLAIMER

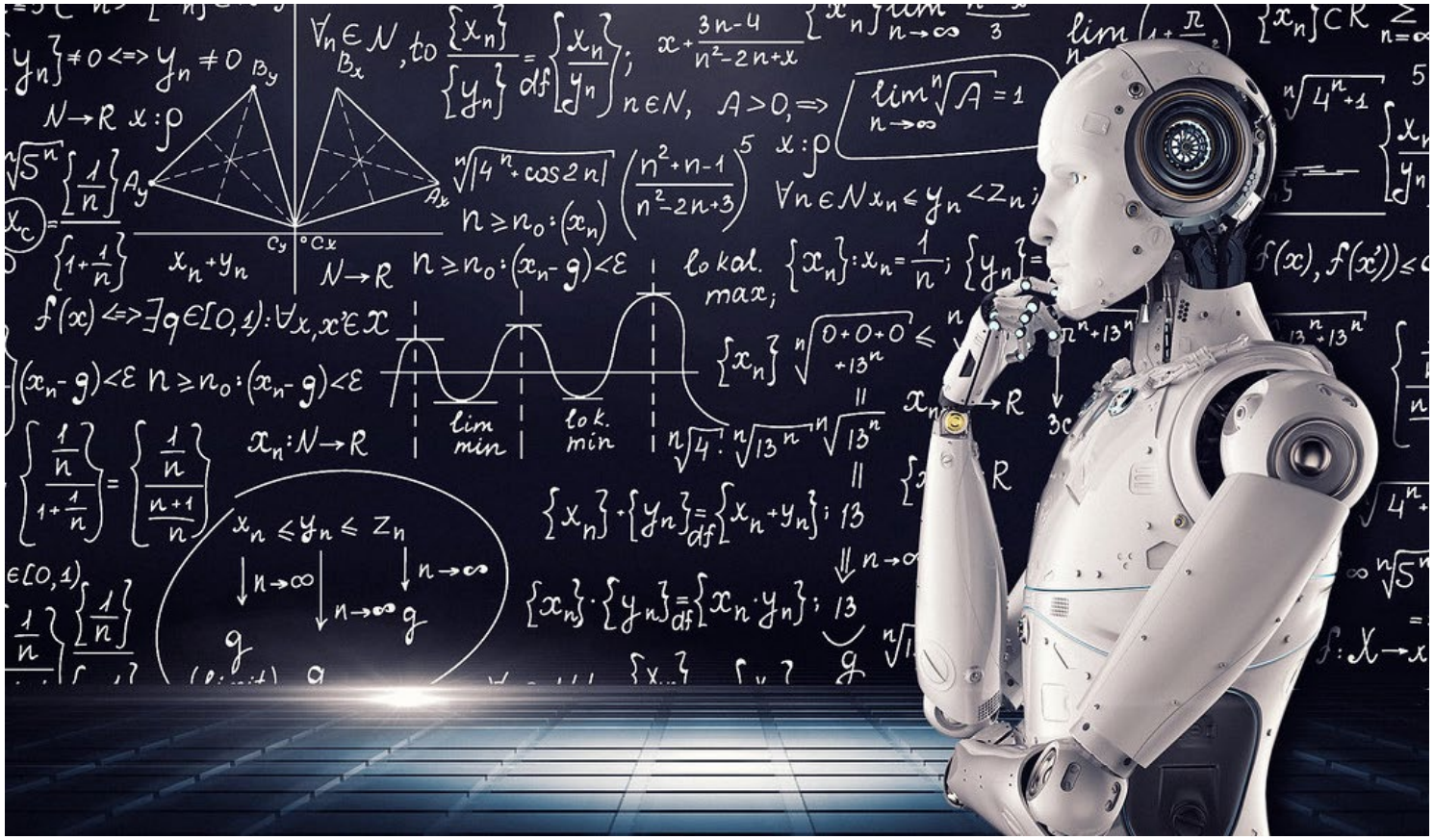
OUTLINE

1. Introduction
2. ECAD Software
3. ECAD Hardware
4. Experiments and Conclusions

PROBLEM STATEMENT

- Designing performant neural networks is difficult
  - Many hyperparameters to consider
  - It is not an exact science
- Fitting optimized neural networks onto hardware is difficult
  - General purpose not difficult and not optimal
  - Creating specialized hardware for a neural network is difficult
    - It is also mostly not feasible



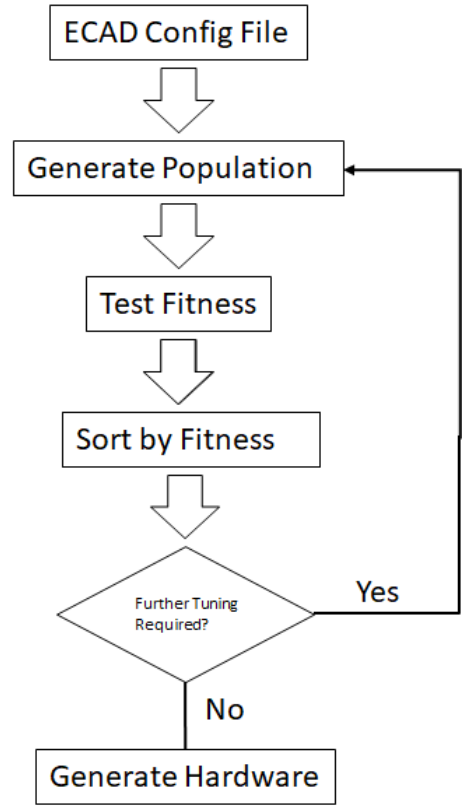


CURRENT SOLUTIONS

- *Artificial* Neural Network Architecture Search (NAS)
  - Random search, evolutionary, reinforcement learning, Bayesian optimizations
- Automated Machine Learning (AutoML)
  - Make machine learning easy for non experts
  - Faster solutions
  - Better solutions
- But these methods still come with problems of their own…
- Few consider acceleration

EVOLUTIONARY CELL AIDED DESIGN (ECAD)

- Artificial neural network and hardware co-optimization
- Configuration file
  - Initial/Base Design
  - Parameters to control the hardware target, objectives and flow
- Evolutionary based design process
  1. Generate Population
  2. Test fitness
  3. Sort by fitness
  - Repeat steps 1-3 until objectives are met
  - Generate hardware

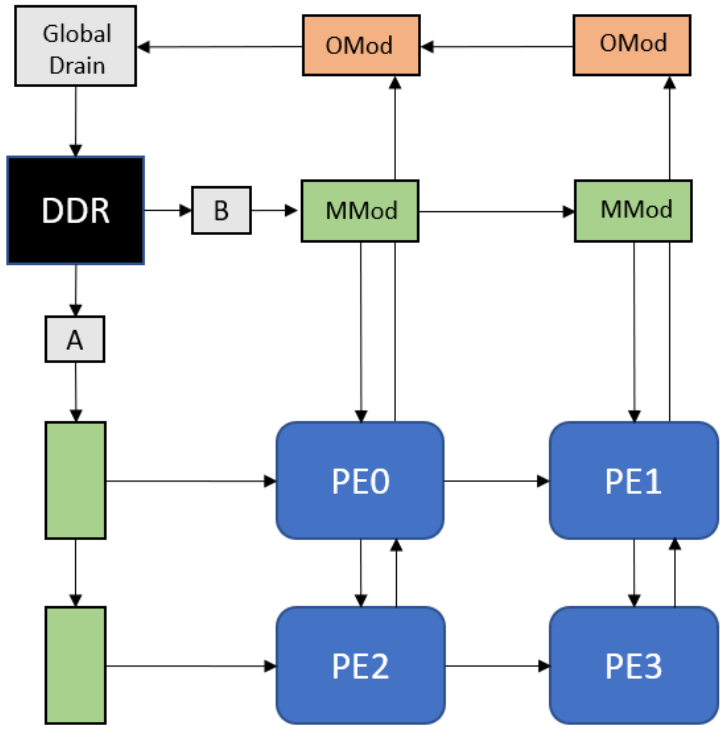


ECAD SOFTWARE

- Auto Generating NN Designs
  - Initially based on configuration file
  - Evolve designs according to fitness (repeatedly apply mutations to best performers)
- Evaluating Population Fitness
  - Simulation Worker capable of simulating a NN design
  - Physical Worker capable of synthesizing a NN design
  - HWDB Worker capable of accurately estimating synthesis results
  - Multi-objective optimization based on the specified goals (accuracy/throughput etc.)
  - Parallel implementation based on MPI sends NN models to workers for evaluation

ECAD HARDWARE FLOW

- ECAD configuration file provides input for accelerator description
  - Memory banks (capacity, bandwidth)
  - FPGA (ALMs, M20K, DSP, F_{max})
- 3 levels for hardware search
  - Software model for fast search, limited resource info
  - Partial compilation, minutes with better resource estimates
  - Full compilation, takes hours for results
- Outputs fitness of permutation
  - Up to the user to provide important metrics



ECAD HARDWARE DESIGN

- Leverages reconfigurability of FPGA
- 2D systolic array
  - Pipelined data flow
  - Scalable and modular
  - Accurate modeling
- Evolving parameters
  - Rows, columns, vector width, interleaving, and scaling
- Hardware fitness
  - Resource utilization, latency, img/s, effective operations per second.

EXPERIMENTS

- Arria 10 GX 1150 FPGA
- Four optimizations considered
  - Accuracy, latency, img/s, and effective gigaoperations/s
- Hardware only search
  - Insights into convergence of hardware designs given optimization pressure
- Hardware and accuracy search
  - Ultimate goal

MODEL ACCURACY

- First step was to verify the model
- Top table shows full compiled results
- Bottom table shows the model results

TABLE III
MEASURED HARDWARE PERFORMANCE RESULTS

HW Config.	Time (ms)	ALM	SRAM	DSP	F_{max} (MHz)
2,8,16,16,2	9.64	37%	36%	26%	220
2,8,32,16,2	5.53	50%	51%	43%	212
4,8,8,16,18	4.65	38%	34%	26%	228

TABLE IV
MODELED HARDWARE PERFORMANCE RESULTS

HW Config.	Time (ms)	ALM	SRAM	DSP	F_{max} (MHz)
2,8,16,16,2	9.59	47%	38%	26%	220
2,8,32,16,2	4.64	61%	55%	43%	212
4,8,8,16,18	4.66	45%	37%	26%	228

UMASS LOWELL

TABLE I
EA RESULTS FOR HARDWARE ONLY OPTIMIZATIONS USING SINGLE AND DUAL BANK DDR MEMORY

Optimization	# Banks	Batch Size	Neurons	BW bound	Latency (ms)	Img/s	EGOP/s	Rows	Cols	Vec	Interleave	Scale
Img/s	1	992	48; 180	Yes	0.068352	1,243,182.00	119	4	8	16	8	2
Latency	1	2	18; 142	No	0.01792	111,607.00	4.0375	2	4	8	4	2
EGOP/s	1	958	992; 1010	No	3.6495	47,508.98	170	4	16	8	16	30
Img/s	2	992	62; 190	No	0.06348	1,532,832.00	191	4	8	16	8	4
Latency	2	158	2; 64	No	0.008096	505,425.00	2.36	2	2	4	2	2
EGOP/s	2	960	998; 1014	No	2.765	49,792.86	179	4	16	8	16	32

HARDWARE ONLY SEARCH

- Img/s provide highest performing throughput designs
- EA resulted in smaller hardware designs, 2 DDR banks had best result
- EGOP/s results in better utilization of resources

TABLE II
TOP PERMUTATIONS FROM OPTIMIZING ACCURACY AND IMG/S

Top Permutation	Fitness Sum	Accuracy	Batch size	Neurons	HW Config.	Latency (ms)	Img/s	EGOP/s
Fitness Sum	1.065	0.936	508	852	2,8,32,16,2	0.352	129,056	174.61
Accuracy	1.006	0.942	484	1018	2,8,16,16,2	0.583	57,296	92.62
Latency (ms)	1.065	0.936	508	852	2,8,32,16,2	0.352	129,056	174.61
Img/s	1.064	0.935	572	970	2,16,32,32,2	1.283	129,144	198.93
EGOP/s	1.062	0.933	572	980	2,16,32,32,2	1.283	129,144	200.98

HARDWARE AND ACCURACY SEARCH

- Fitness sum provides the weighted accumulation of accuracy and img/s
- Fitness sum and latency arrived at the same design
  - Table I showed Latency reducing number of neurons so with accuracy combating this, it resulted in a good design

THANK YOU