Imperial College
London

Algorithm and Hardware Co-design for Reconfigurable CNN Accelerator

Hongxiang Fan, Martin Ferianc, Zhiqiang Que, He Li, Shuanglong Liu, Xinyu Niu and Wayne Luk
Department of Computing, Imperial College London
h.fan17@imperial.ac.uk

Motivation

- Neural architecture search (NAS):
  - Algorithm-level optimization: better neural architectures
- Reconfigurable accelerator
  - Hardware-level optimization: higher hardware efficiency
- Suboptimality when both techniques are not considered jointly
  - Design of neural architecture space needs to consider the characteristics of hardware accelerators.

Challenges

- Large co-design space
  - Millions of design points in ResNet-101

- Expensive training cost
  - Several hours for large neural networks (ResNet-101) and datasets (ImageNet) even on high-end GPUs

- Time-consuming hardware implementation
  - Runtime cost of synthesis and place&route is expensive

Contributions

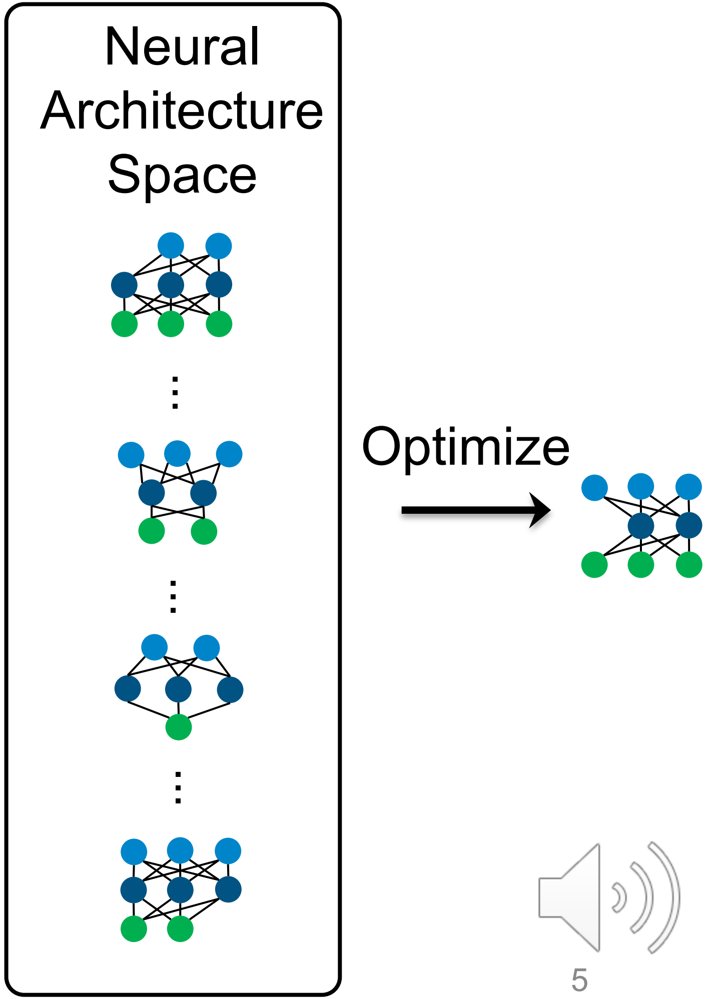
1. A novel co-design framework for reconfigurable CNN designs
   - Evolutionary algorithm for design space exploration
   - Decoupling network training from design optimization process to reduce training cost

2. Gaussian process (GP)-based estimators
   - Avoid time-consuming synthesis and place&route

3. Comprehensive comparisons
   - with various co-design methods

Background: NAS

- Auto machine learning: Optimize hyperparameters of neural architecture:
  - Number of channels
  - Number of layers
  - Operations
  - Connection between layers



Neural Architecture Space

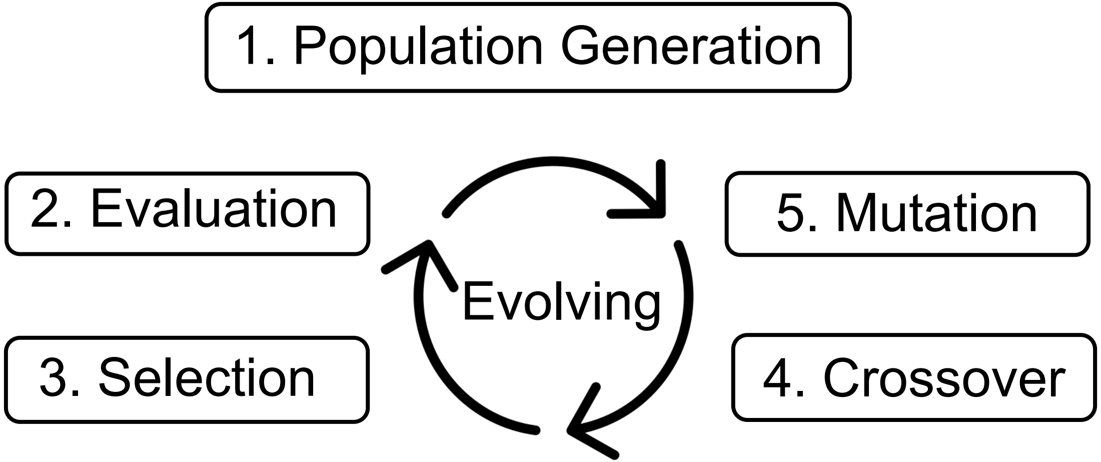
Optimize

Background: Gaussian Process

- A supervised learning approach
  - Regression
  - Probabilistic classification (confidence intervals)
- A generalization of the Gaussian probability
  - Distribution of functions instead of vectors
- Parameterized by
  - Mean function: Usually set as zero
  - Covariance function: Defining the prior properties of the functions

Background: Evolutionary Algorithm

- Population-based optimization algorithm inspired by biological evolution
- Contain five stages for iterative optimization
- Parameterized by population size, parent size, mutate probability and cross over probability



1. Population Generation

2. Evaluation

3. Selection

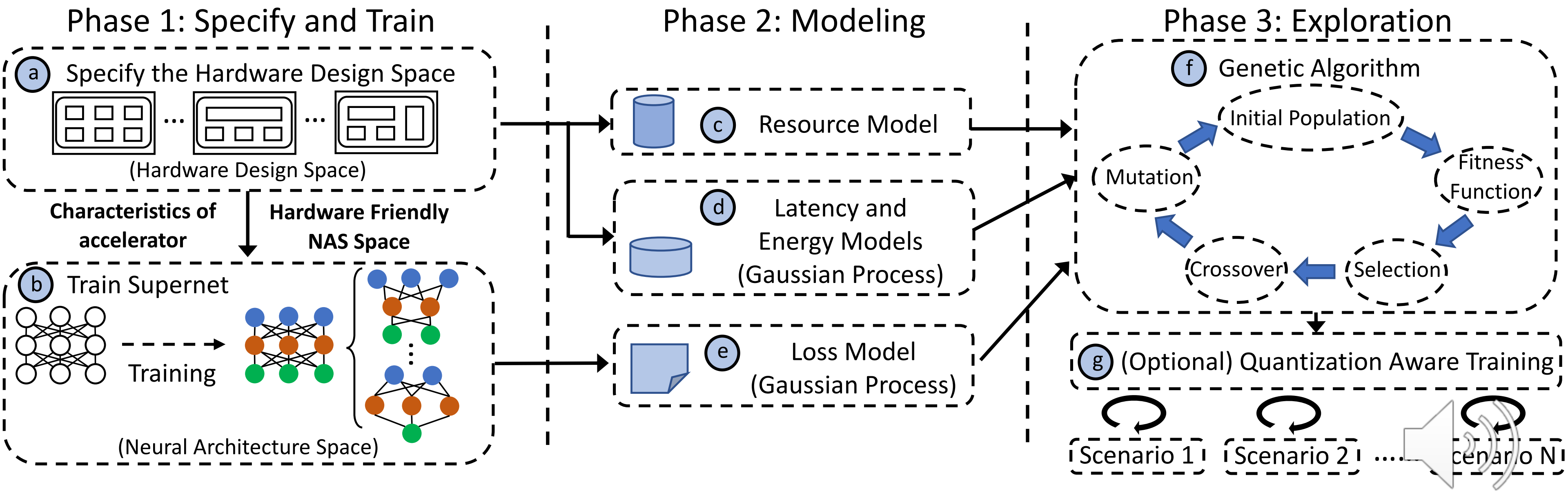
4. Crossover

5. Mutation

Evolving

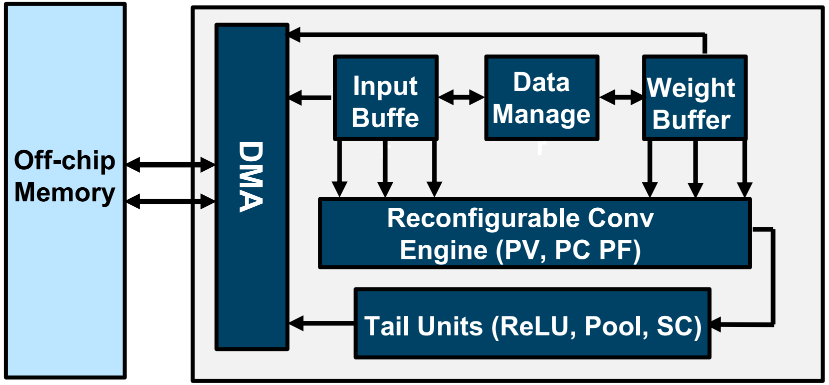
Co-Design Framework: Overview

- Three-phase co-design framework
- General enough for any reconfigurable CNN accelerators


Phase 1: Specify and Train
Phase 2: Modeling
Phase 3: Exploration
a
Specify the Hardware Design Space
(Hardware Design Space)
Characteristics of accelerator
Hardware Friendly NAS Space
b
Train Supernet
Training
(Neural Architecture Space)
c
Resource Model
d
Latency and Energy Models (Gaussian Process)
e
Loss Model (Gaussian Process)
f
Genetic Algorithm
Initial Population
Fitness Function
Selection
Crossover
Mutation
g
(Optional) Quantization Aware Training
Scenario 1
Scenario 2
Scenario N

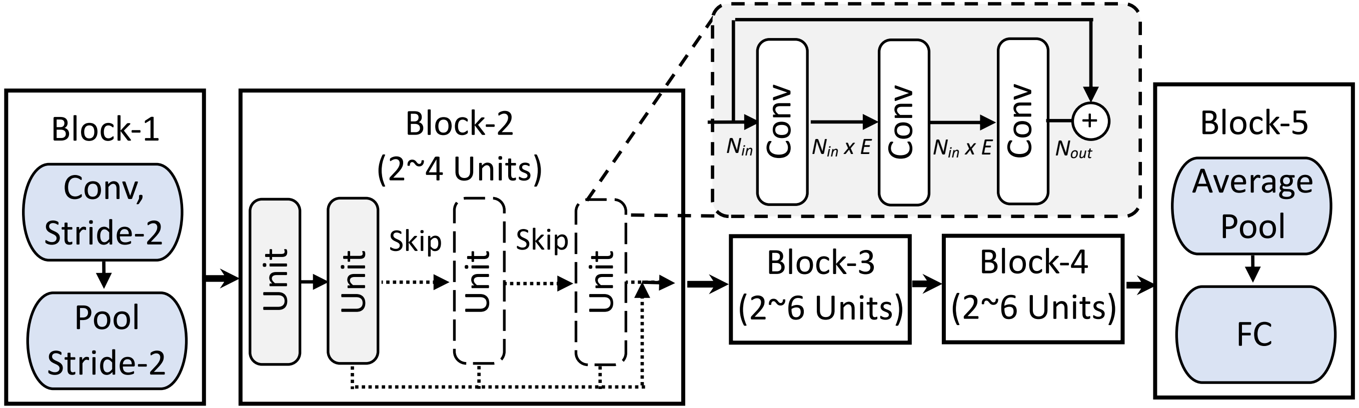
Co-Design Framework: Phase One

- Hardware Architecture Space
  - One single reconfigurable engine for different layers
  - Off-chip memory to cache intermediate results
  - Reconfigurable parameters:
    - Parallelism along channel (PC), filter(PF) and vector (PV)
    - Memory size
    - Bandwidth


Off-chip Memory
DMA
Input Buffe
Data Manage
Weight Buffer
Reconfigurable Conv Engine (PV, PC PF)
Tail Units (ReLU, Pool, SC)

Co-Design Framework: Phase One

- Neural Architecture Space (Supernet)
  - ResNet Backbone with five blocks
  - Optimize: number of units, number of channels and connections between layers
  - Train supernet using progressive shrinking



Co-Design Framework: Phase Two

- Estimation for training loss
  - Encode neural architectures (channel number, filter number…..) as vectors for numerical analysis

- Estimation for latency and energy consumption
  - Encode neural architecture as well as hardware parameters (parallelism level in different dimensions, buffer sizes and bandwidth) for regression

Co-Design Framework: Phase Three

- Explore co-design space
  - Evolutionary algorithm
  - Represent design points using encoded genes
  - Flexibility: User-defined parameters

- Quantization
  - Reduce the bandwidth and memory requirements
  - Optional according to the availability in hardware

Co-Design Framework: Phase Three

- Trade-off between algorithmic and hardware performance
  - Parameterized objective function
  - Set different optimization priorities
  - Penalty: resource usage exceeds the budget

$$\mathcal{L} = \eta \times CE + \mu \times Latency + \lambda \times Energy + Res_{PT}$$

$$Res_{PT} = \begin{cases} 0, & DSP_{used} \leq DSP_{avl}, MEM_{used} \leq MEM_{avl} \\ \gamma, & DSP_{used} > DSP_{avl}, MEM_{used} > MEM_{avl} \end{cases}$$

Experiment: Setup

- Hardware implementation
  - Intel Arria 10SX 660 platform
  - Verilog HDL, 225 MHz

- Optimization Framework
  - Python 3.6, PyTorch 1.9 and Gpytorch 1.5

- Image classification on ImageNet
  - 1000 objects

Experiment: Accuracy of GP-based Estimators

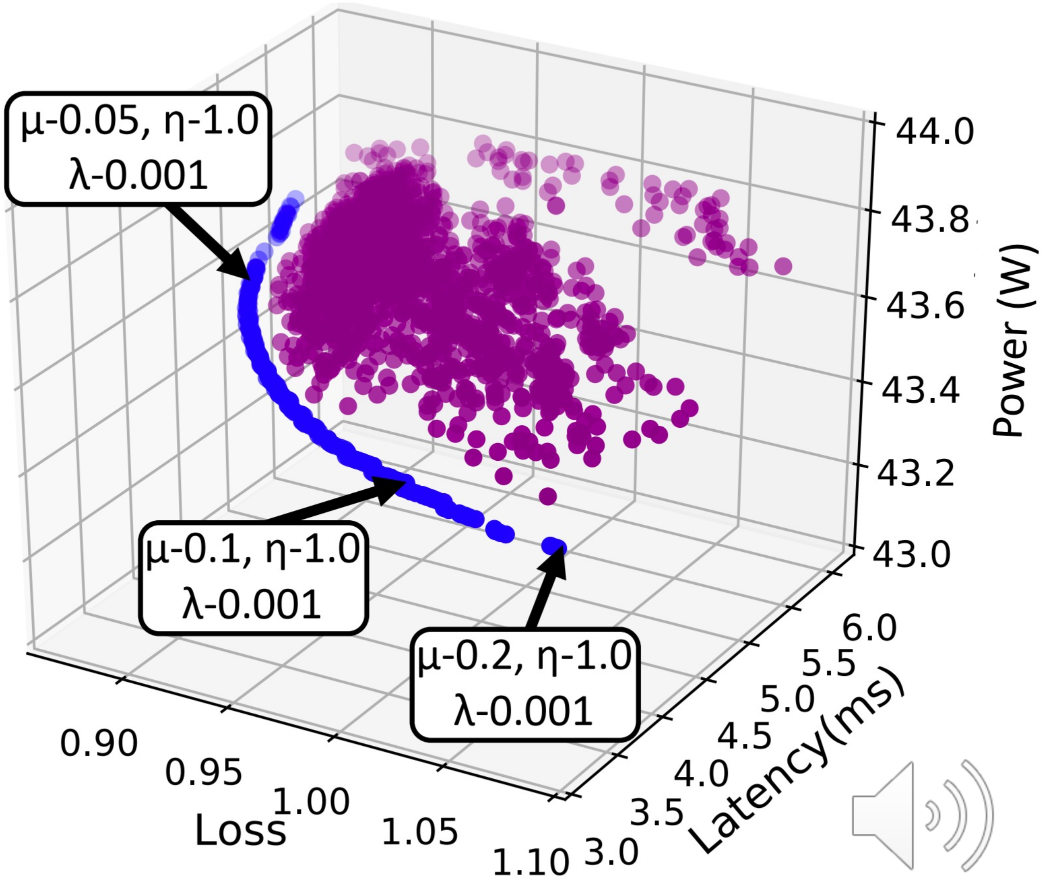
- 1500 samples for training, 500 samples for evaluation
- Adam optimizer, 50 iterations

	Kernel Function	Mean Absolute Error
Loss Estimator	Matern (3/2)	0.01005
Latency Estimator	Matern (5/2)	0.06521 ms
Energy Estimator	Matern (5/2)	0.01804 W

Experiment: Effectiveness of Co-design Framework

- Three different optimization settings
  - Opt-Accuracy
  - Opt-Power
  - Opt-Latency

- Effectively find the Pareto frontier



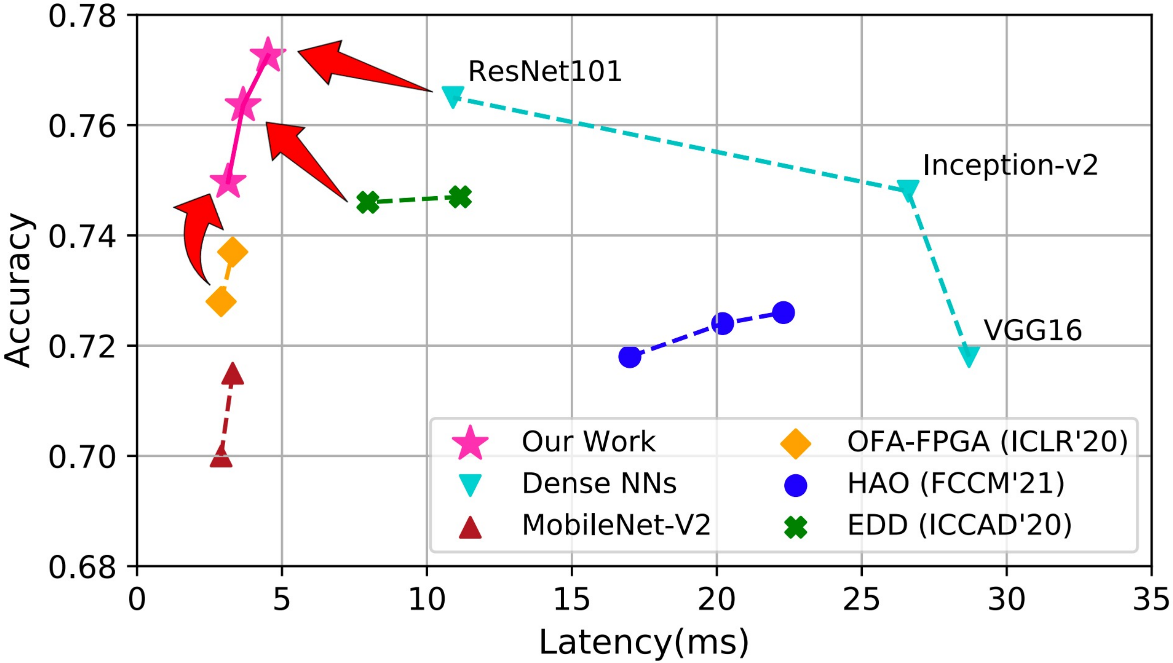
Experiment: Comparison with CPU and GPU

- Higher energy efficiency and lower latency

	CPU		GPU		FPGA		Acc (%)
	Latency (ms)	Energy Eff. (FPS/W)	Latency (ms)	Energy Eff. (FPS/W)	Latency (ms)	Energy Eff. (FPS/W)	
Opt Acc	26.08	0.28	7.40	0.94	4.52	5.07	77.63
Opt Energy	24.06	0.30	6.57	1.06	4.66	6.27	76.30
Opt Lat	19.18	0.38	5.03	1.38	3.14	7.32	74.91

Experiment: Comparison with Handcrafted CNNs

- Higher accuracy under same latency constraint
- Faster under same accuracy constraint



Experiment: Comparison with Other Approaches

- Higher accuracy
- Higher energy efficiency

	Platform	# of DSP	Latency (ms)	Accuracy	Energy Eff. (GOPS/W)
Co-Explore	Xilinx XC7Z015	150	95.24	70.24%	0.74
HAO	Xilinx ZU3EG	360	22.27	72.68%	-
EDD	Xilinx ZCU102	2520	7.96	74.60%	-
OFA	Xilinx ZU9EG	2520	3.30	73.60%	-
Our Work	Intel GX1150	1345	3.66	76.30%	6.27

Future Work

- Extend the framework to support other DNNs
  - Bayesian neural networks, Recurrent neural networks

- Improve the hardware architecture to support more operations
  - More general and efficient

- Automatic sparsity exploitation
  - Structured and unstructured sparsity

Summary

1. A novel co-design framework for reconfigurable CNN designs
   - Evolutionary algorithm for design space exploration
   - Decoupling network training from co-design optimization process to reduce training cost

2. Gaussian process (GP)-based estimators
   - Avoid time-consuming synthesis and place&route

3. Comprehensive comparisons
   - Higher accuracy, Lower latency, Higher energy-efficiency