



Cost-effective Hardware Accelerator Recommendation for Edge Computing

Xingyu Zhou, Robert Canady, Shunxing Bao, Aniruddha Gokhale
DOC-VU Group, Dept of EECS
Vanderbilt University, Nashville, TN 37235



Outline



- Current Edge HW Acc Status
- Challenge for HW Acc Deployment
- Solution Overview
- Case Study
- Conclusion





What are HW Accelerators?



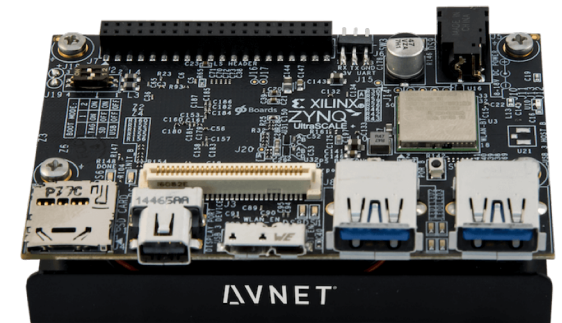
- Accelerating computations
- For general or specific task settings

CPU (most general)

GPU (better suited for stream processing)

FPGA (general in theory but difficult to use)

ASIC (specific)

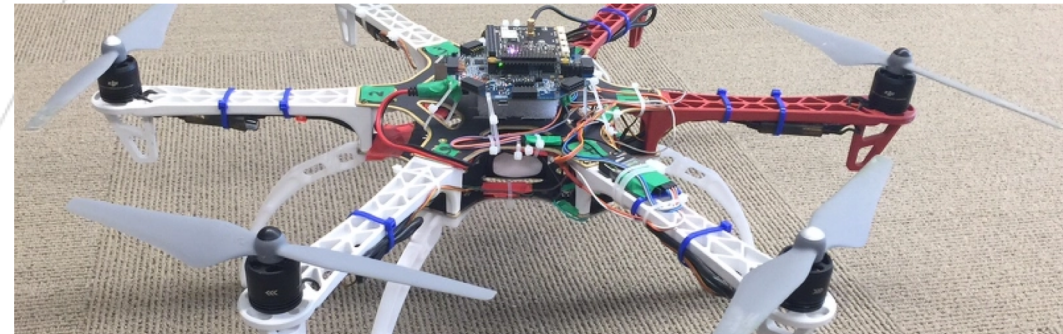




Why Hardware Accelerators on Edge?



- Heterogeneous data sources from sensors;
- More compute intense processing requirements especially from image or video;
- Realistic physical constraints(power,size,cost. etc)

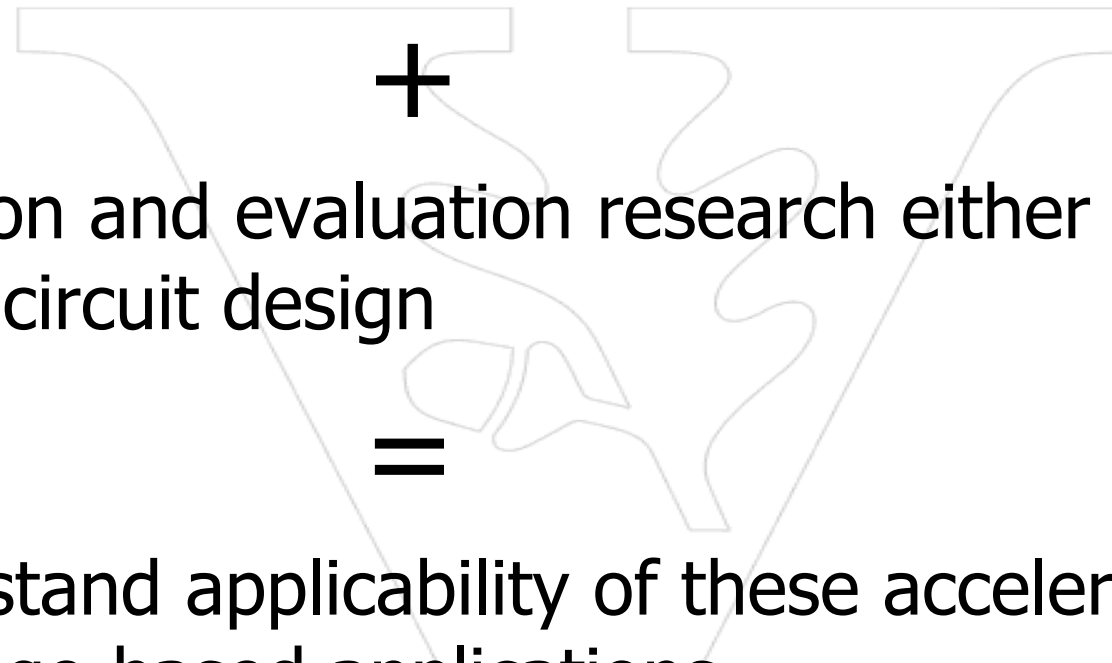




Challenge: which accelerator is best suited for application needs?



- Too many different hardware devices potential for edge



- Current selection and evaluation research either single device or even low-level circuit design
- Need to understand applicability of these accelerator technologies for at-scale, edge-based applications



Metrics for HW Acceleration Evaluation

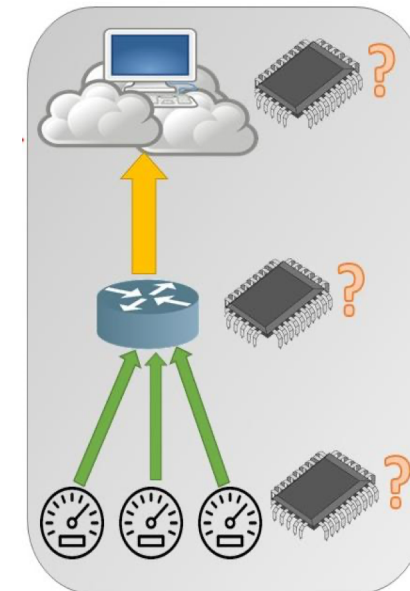


- Latency => Application Response
- Power => Electricity Cost
- Commercial Cost => Market Price

MAO *et al.*: SURVEY ON MOBILE EDGE COMPUTING: COMMUNICATION PERSPECTIVE

TABLE III
CHARACTERISTICS OF TYPICAL WIRELESS COMMUNICATION TECHNOLOGIES

	NFC	RFID	Bluetooth	WiFi	LTE	5G
Max. Coverage	10cm	3m	100m	100m	up to 5km	Excellent coverage
Operation Freq.	13.56MHz	LF: 120-134kHz HF: 13.56MHz UHF: 850-960MHz	2.4GHz	2.4GHz, 5GHz	TDD: 1.85-3.8GHz FDD: 0.7-2.6GHz	6-100GHz
Data Rate	106, 212, 414kbps	Low (LF) to high (UHF)	22Mbps	135Mbps (IEEE 802.11n)	DL: 300Mbps UL: 75Mbps	Indoor/dense outdoor: up to 10Gbps Urban/suburban: > hundreds of Mbps



V. Sze, T.-J. Yang, Y.-H. Chen, J. Emer, "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295-2329, December 2017.



Overall Goal for HW Selection



- Define One HW Acceleration Strategy:
 - HW Acceleration Task Realization on Device
 - HW Acceleration Device Placement (location,time)

- Minimize deployment cost under constraints

Current goal: minimize cost with design latency limit

$$\min_{dev \in ListHW} \sum costHW_{dev} * nHW_{dev} + costP(dev, T_{cycle})$$

subject to:

$$T_{app}(dev) \leq t_{target}$$

$$costP(dev, T_{cycle}) = P_{app}(dev, T_{cycle}) * costElec$$

$$P_{app}(dev, T_{cycle}) = P_{idle}(dev) * T_{cycle} + P_{perinf}(dev) * muFreq_{in} * T_{cycle}$$



Cost Evaluation Workflow Part I



1. Application design

choose applications that can be accelerated

ResNet50 (Classification) + TinyYolo (Detection)

2. Hardware configuration

go through design flows

Table 1: Device-level Acceleration Deployment Workflows for Different Hardware Platforms

Design Flow	Edge CPU	Embedded GPU	FPGA	ASIC	Server GPU	Server CPU
Hardware	Raspberry Pi 3 b+	NVIDIA Jetson Nano	Avnet Ultra96	Intel NCS	NVIDIA GTX1060 6Gb	AMD FX-6300
ResNet-50	Tensorflow/Keras	TensorRT	DNNDK	OpenVINO	Tensorflow/Keras/Cuda	Tensorflow/Keras
Tiny Yolo	Darknet	Darknet/TensorRT	DNNDK	OpenVINO	Tensorflow/Keras/Cuda	Tensorflow/Keras



Cost Evaluation Workflow Part II



3. Per-Device Benchmarking

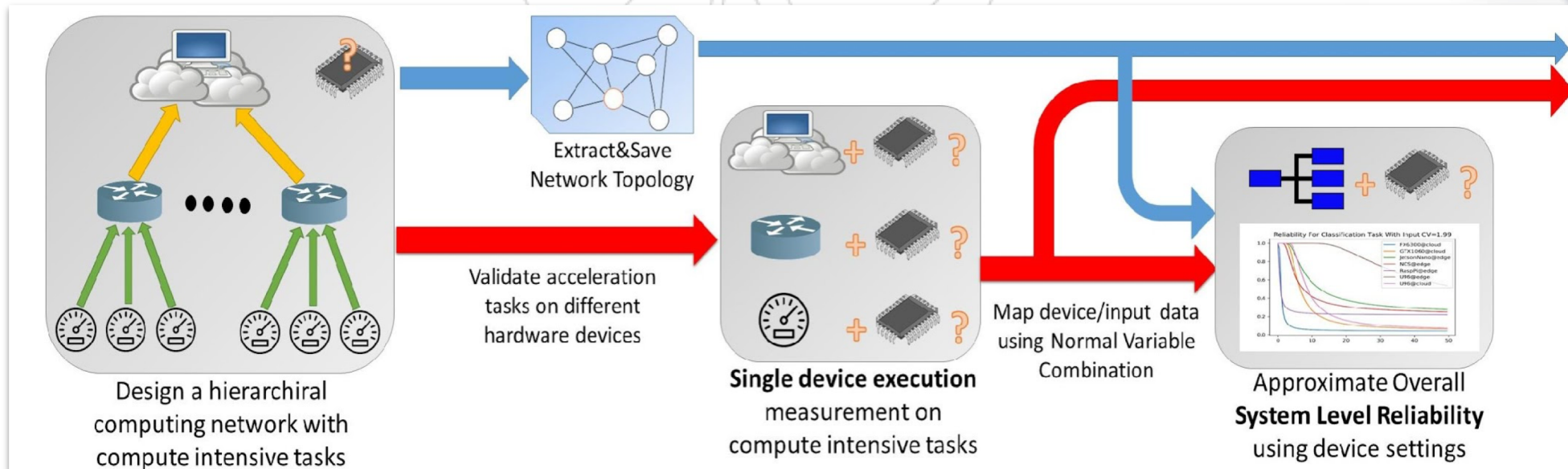
record time and power consumption

4. Deployment Cost Approximation

= devCost (hardware market price)

+ deployCost (for design topology and time cycle)

5. Choose device met requirements





Per-device Applicability Validation



Applicability Test on Relative High Dimension Data:

Object Classification tasks on a set of 500 images with a resolution of $640 * 480$.

Vehicle Detection tasks on a road traffic video consisting of 874 frames with a resolution of $1280 * 720$.

Table 2: Response Time (T_{hw}) for Object classification Task using *ResNet-50* (Unit: Second)

Time	RPi	JetsonNano	Ultra96	NCS	GTX1060	FX6300
mean	2.089	0.133	0.029	0.218	0.039	0.268
std	0.058	0.016	0.001	0.003	0.005	0.006

Table 3: Power Consumption for Object classification using *ResNet-50* (Unit: Watt)

Power	RPi	JetsonNano	Ultra96	NCS	GTX1060	FX6300
Idle	1.8	2.2	6.2	0.4	10	72
Infer	4.8	5.6	7.6	1.9	122	145

Table 4: Response Time (T_{hw}) for Traffic Detection Task using *Tiny Yolo* (Unit: Second)

Time	RPi	JetsonNano	Ultra96	NCS	GTX1060	FX6300
mean	2.874	0.096	0.023	0.238	0.059	0.217
std	0.068	0.008	0.001	0.003	0.002	0.076

Table 5: Power Consumption for Traffic Detection using *Tiny Yolo* (Unit: Watt)

Power	RPi	JetsonNano	Ultra96	NCS	GTX1060	FX6300
Idle	1.8	2.3	7.4	0.4	10	72
Infer	4.8	11.7	9.2	2.1	122	150



At-Scale Approximation

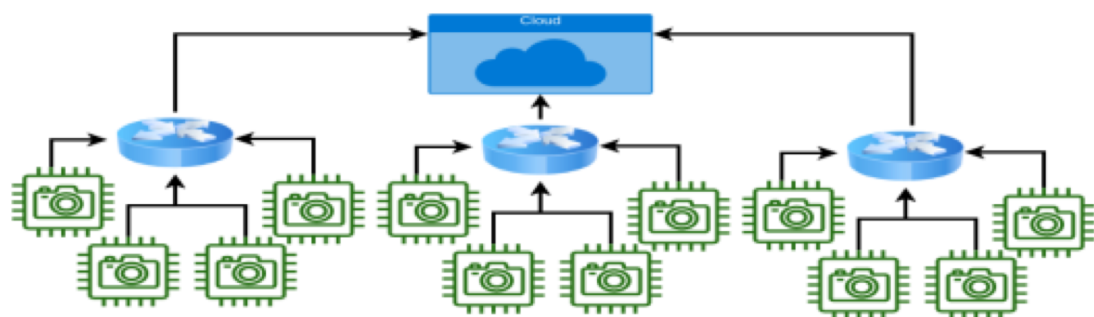


Figure 1: Three-level Design Topology Layout: (1) Top:Cloud servers; (2) Intermediate:3 Fog groups include communication control and some computation power; (3) Bottom:4 Edge nodes in each fog group closest to sensors and data needs to be processed.

$$R_{\text{dev}} \sim N(\mu\text{Freq}_{\text{dev}} * n\text{HW}_{\text{dev}}, \text{stdFreq}_{\text{dev}}^2)$$

$$L_{\text{dev}} \sim N(\mu\text{Freq}_{\text{in}}, \text{stdFreq}_{\text{in}}^2)$$

$$\Pr(R_{\text{dev}} - L_{\text{dev}}) > \text{conf}$$

Design Topology Potential Scenarios:

1. unmanned shopping using object classification
2. surveillance using detection

Reliability-Driven System Deployment Goal:

1. should guarantee to handle no less than half (2 of 4) of input loads from every fog group (3 groups) with an overall confidence level of 99%
2. edge node inputs denoted by a normal distribution (assumed identical for all nodes in this topology)
3. edge node inputs with relatively high uncertainty level with $\text{stdFreq}_{\text{in}} = \mu\text{Freq}_{\text{in}}$ (inputCV=1.0)



At-Scale Approximation



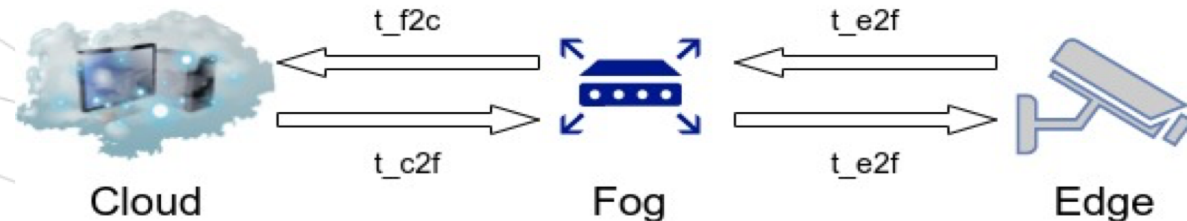
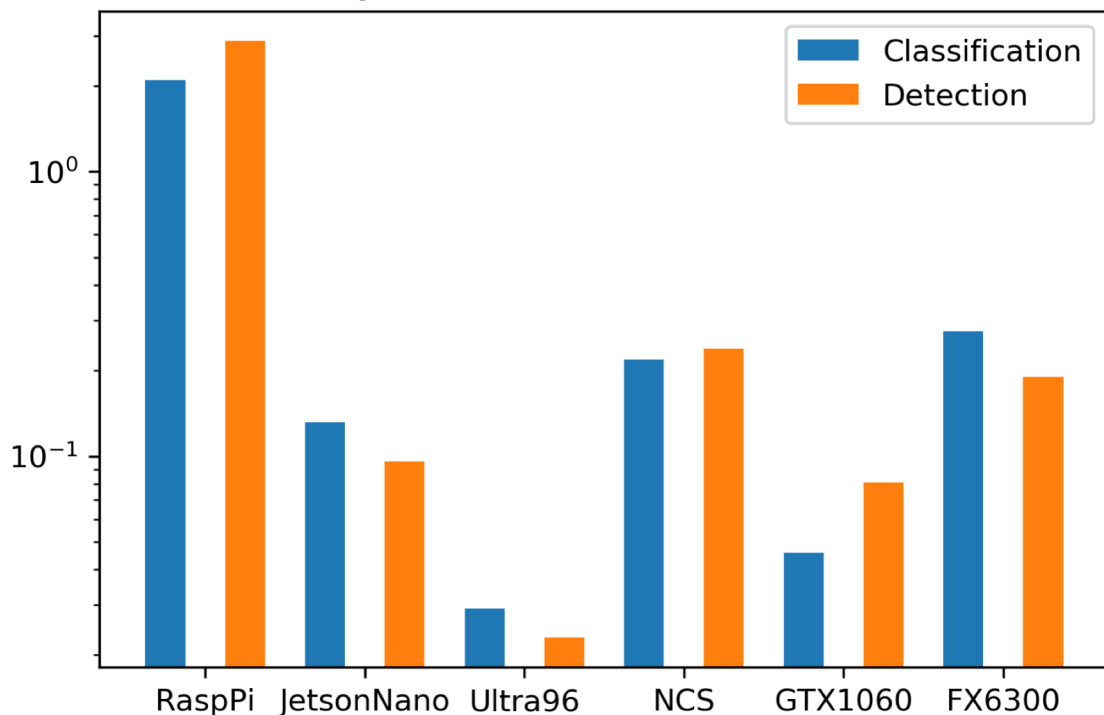
Table 2: Response Time (T_{hw}) for Object classification Task using *ResNet-50* (Unit: Second)

Time	RPi	JetsonNano	Ultra96	NCS	GTX1060	FX6300
mean	2.089	0.133	0.029	0.218	0.039	0.268
std	0.058	0.016	0.001	0.003	0.005	0.006

Table 4: Response Time (T_{hw}) for Traffic Detection Task using *Tiny Yolo* (Unit: Second)

Time	RPi	JetsonNano	Ultra96	NCS	GTX1060	FX6300
mean	2.874	0.096	0.023	0.238	0.059	0.217
std	0.068	0.008	0.001	0.003	0.002	0.076

Latency Per Inference Period for Devices



$$T_{app}(dev) = T_{hw} + T_{comm} = \mu T_{dev} + t_{e2f} + t_{f2c}$$

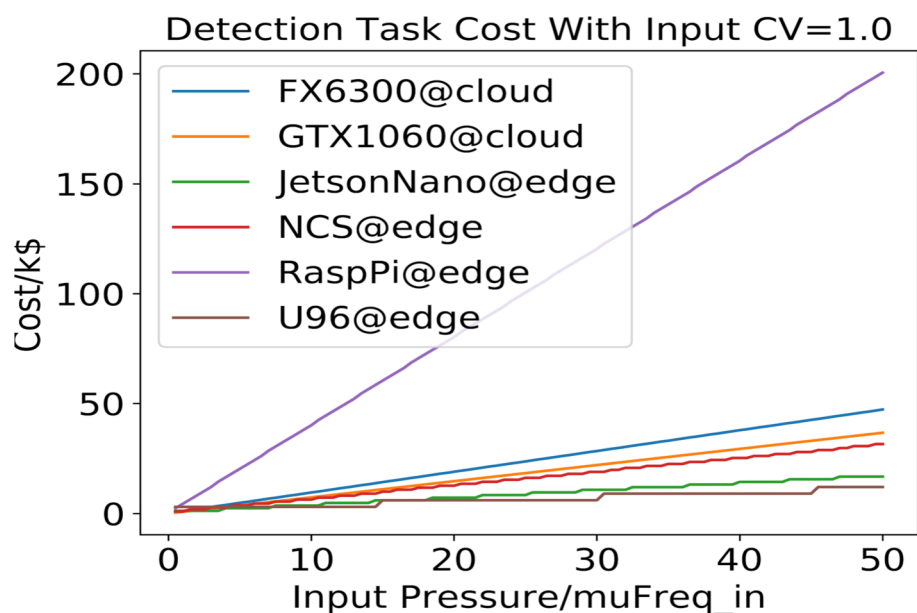
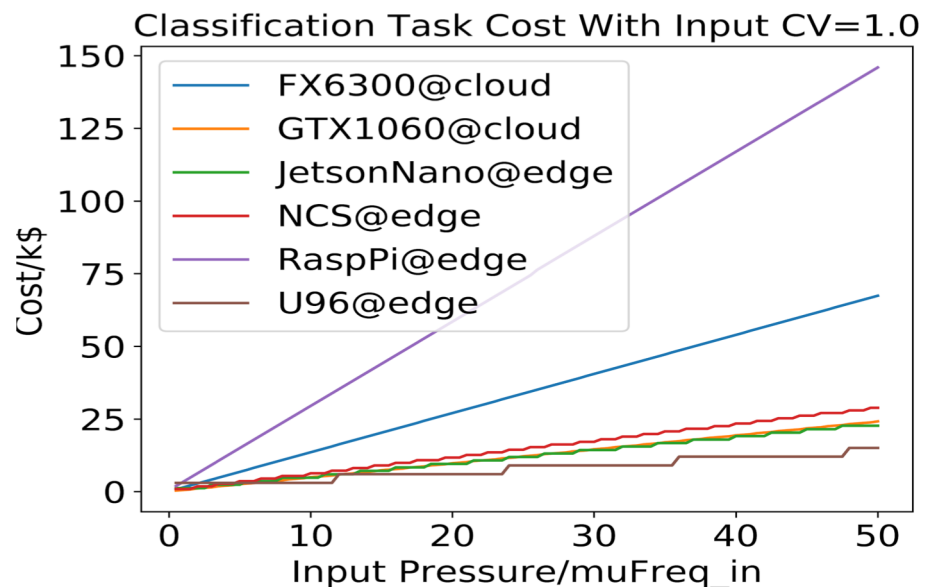
Table 6: Inference Cycle Time $T_{app}(dev)$ (Unit:Second)

Time	RPi	JetNano	Ultra96	NCS	GTX1060	FX6300
Loc	Edge	Edge	Edge	Edge	Cloud	Cloud
Res	2.088	0.131	0.029	0.218	0.046	0.275
Yolo	2.869	0.096	0.023	0.238	0.081	0.190

Bandwidth Setting: standard IEEE802 Wifi with 135Mbps



At-Scale Approximation



Settings:

Increasing input strength for a 24-month deployment cycle


1. Why hardware accelerator necessary?
CPUs: RaspPi@edge, FX6300@cloud worst
2. Power is critical for long-term
two most cost-efficient options for edge:
Ultra96 (FPGA)
Jetson Nano (embedded GPU)
3. Device tradeoff:
FPGAs hard to use, NCS not powerful



Presents a simple evaluation procedure as a recommendation system to help users select an accelerator hardware device for their applications deployed across the cloud to edge spectrum

Cons:

1. A pure strategy of one single type of device is considered
 2. One single type of acceleration task is set for all devices
- Plan to investigate at-scale deployment of RNN and GAN in edge scenarios;
3. Assume an ideal device task scheduling and device parallelism
 4. Have not taken interference effects between device executions into consideration



Thank You!
Q&A