Invocation-driven Neural Approximate Computing with a Multiclass-Classifier and Multiple Approximators

Haiyue Song, Chengwen Xu, Qiang Xu, **Zhuoran Song**, Naifeng Jing, Xiaoyao Liang, and Li Jiang
Advanced Computer Architecture Laboratory
Shanghai Jiao Tong University

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

1 Background

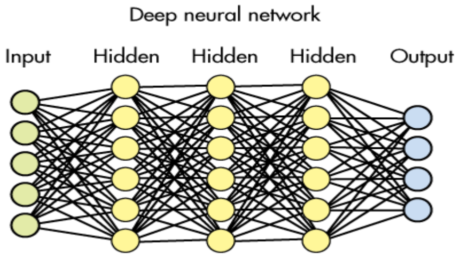
2 Related works and Motivation

3 Proposed Method

4 Experiment Results

5 Conclusion

Approximate Computing



Machine Learning

Robotics

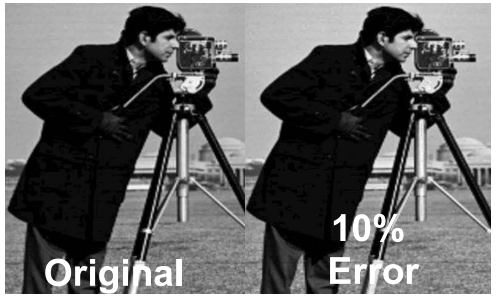


Image Processing

Data Mining

- Many applications are error tolerant
- Neural network (NN) is suitable to approximate a code block/function
  - Amdahl law: performance limited by serial code
  - NN has high parallelism, e.g., FPGA, ASIC, GPU
  - An interesting facts: Neural network can approximate any continuous function

1 Background

2 Related works and Motivation

3 Proposed Method

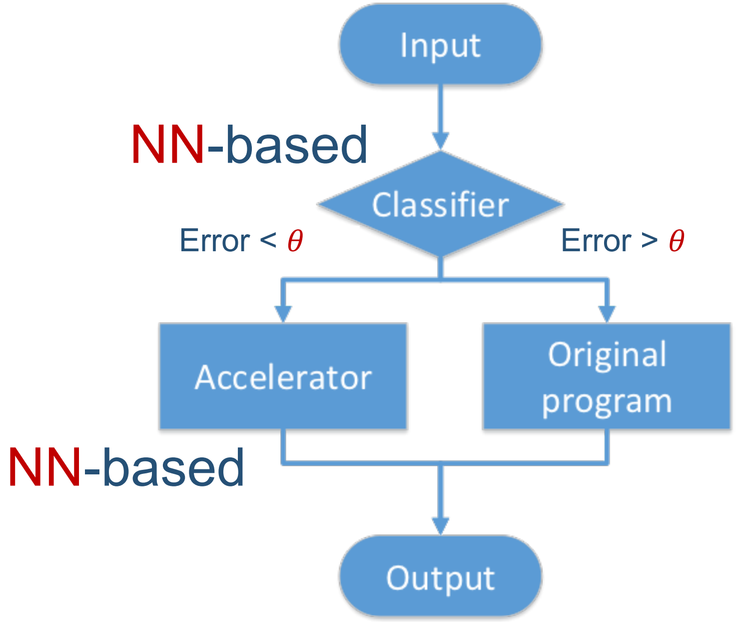
4 Experiment Results

5 Conclusion

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Related works

- **Model based quality control for Approximate Computing [ISCA'15, ISLPED'16, DATE'16]**
  - Classifier：predict the data is "approximatable" or not
  - Approximator (Accelerator)：approximately compute data at **fast** speed and **low** power consumption
  - Error：the gap between the output of approximator and that of original program

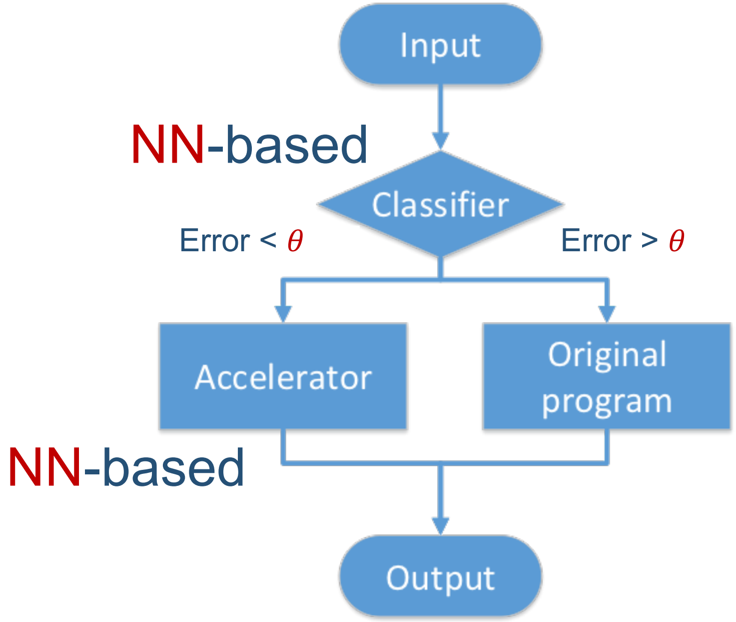


With quality control architecture

Related works

- **Model based quality control for Approximate Computing [ISCA'15, ISLPED'16, DATE'16]**
  - Classifier：predict the data is "approximatable" or not
  - Approximator (Accelerator)：approximately compute data at **fast** speed and **low** power consumption
  - Error：the gap between the output of approximator and that of original program
- **Question:**

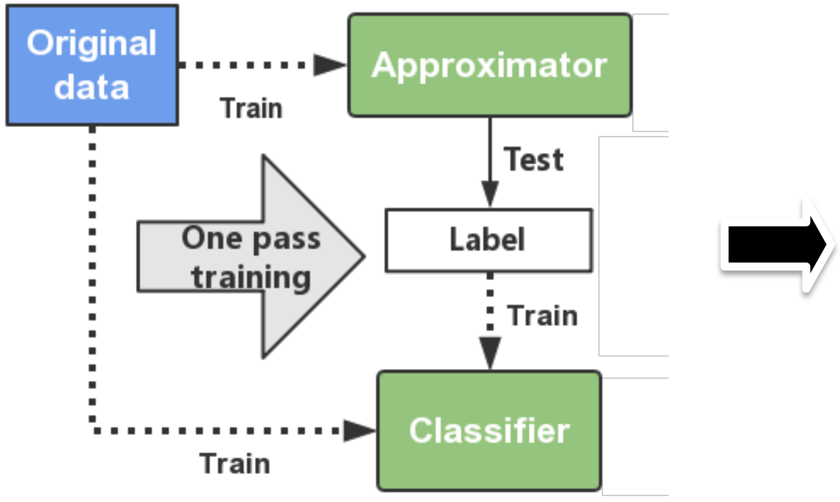
How to train NN-based classifier and approximator?

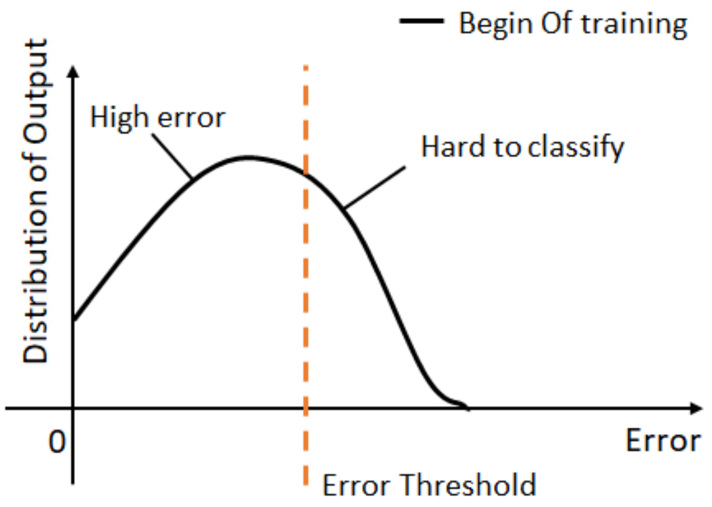


With quality control architecture

Related works

- **One-pass training[ISCA'16]**
  - Train Approximator and Classifier separately
  - Ignore the correlation between the two NNs

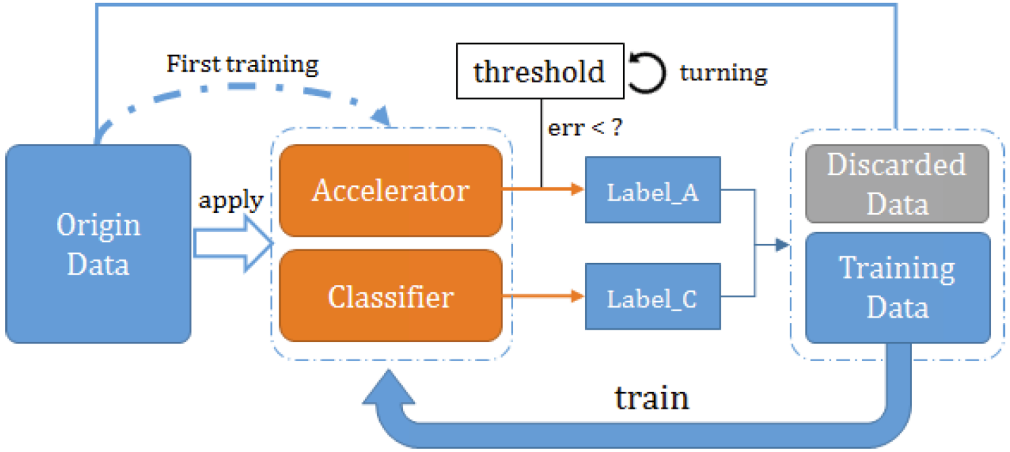


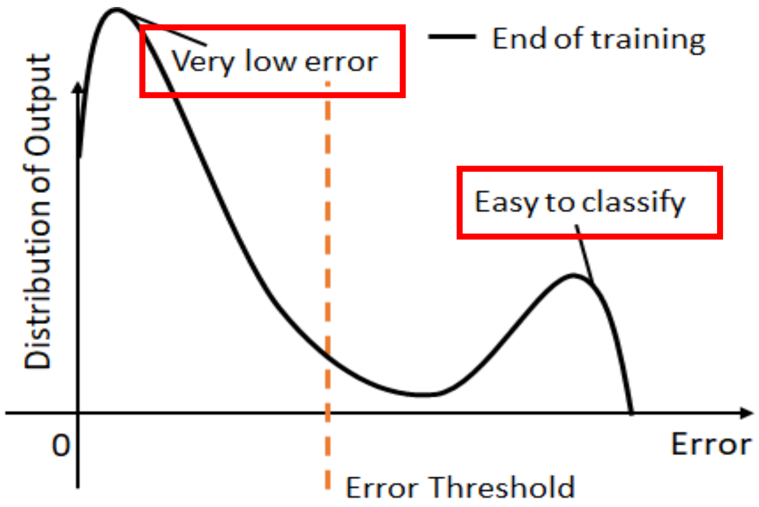


One-pass training method

Related works

- **Iterative training[DAC'17]**
  - Train Approximator and Classifier together using iterative training
  - Classifier correlate with Approximator
  - Data with low error is easy to predict

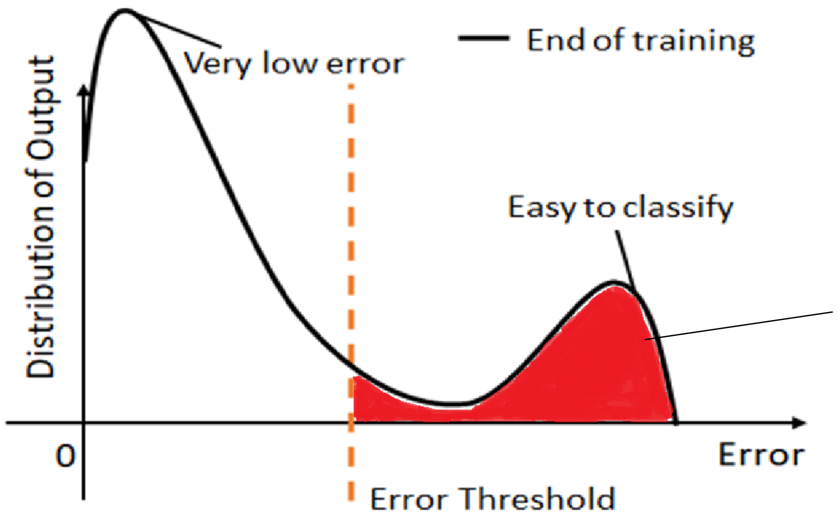




Iterative training

Motivation

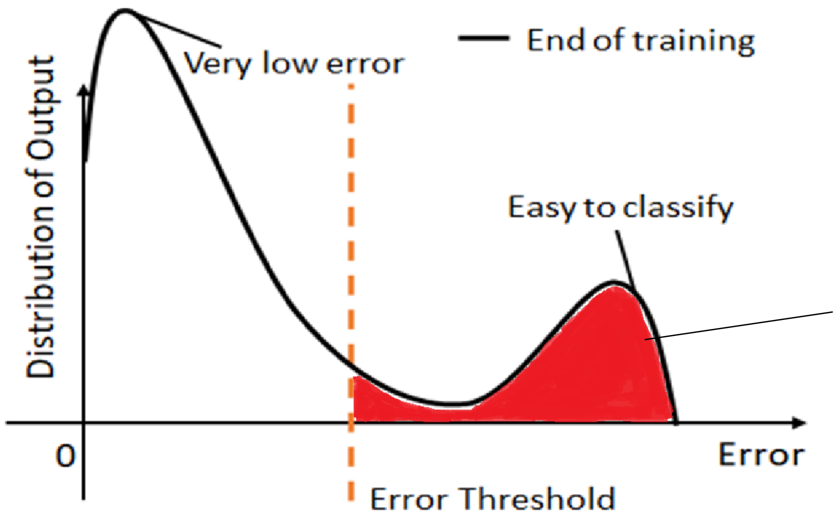
- **Problems**
  - Even iterative training, some data still fail to be approximated (red part in the figure)
  - Single Approximator may overfit one cluster/distribution of input sample



Do we really have to give up those data?

Motivation

- Problems
  - Even iterative training, some data still fail to be approximated (red part in the figure)
  - Single Approximator may overfit one cluster/distribution of input sample
- Motivation
  - Multiple approximators may be complementary, and **make invocation higher**



Do we really have to give up those data?

1 Background

2 Related works and Motivation

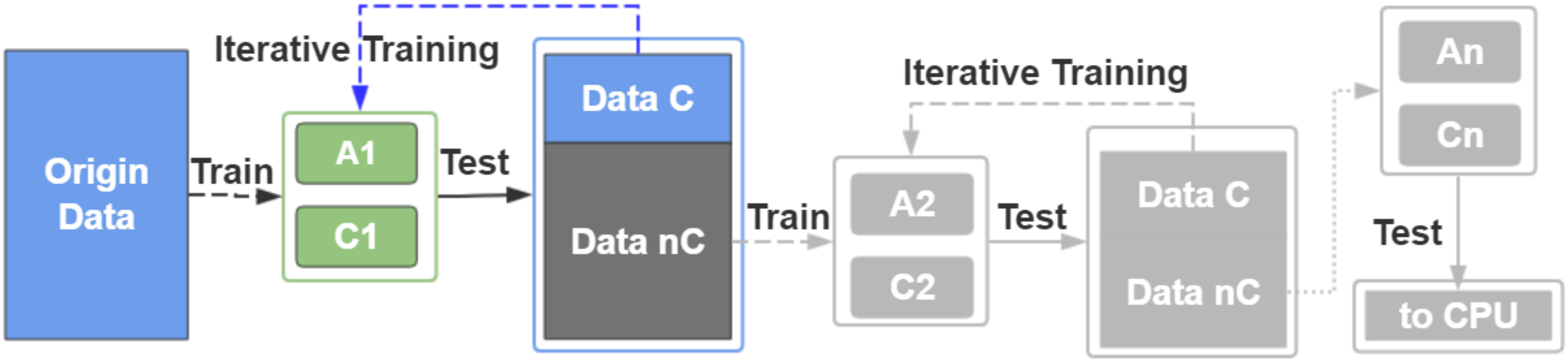
3 Proposed Method

4 Experiment Results

5 Conclusion

Multiple Cascaded Classifiers and Approximators (MCCA)

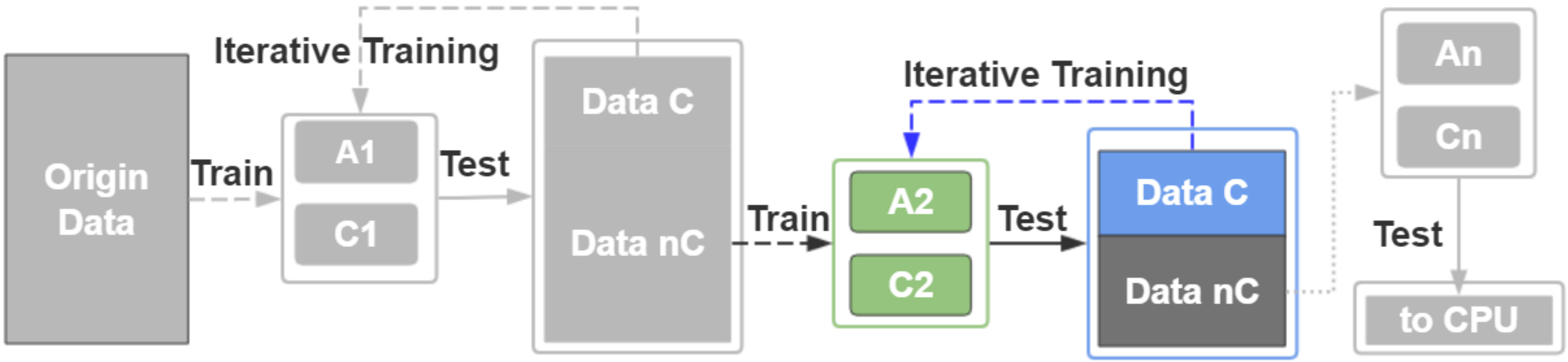
- **Training Process**
  - The original input samples are used to train classifier C1 and approximator A1.



Training Process

Multiple Cascaded Classifiers and Approximators (MCCA)

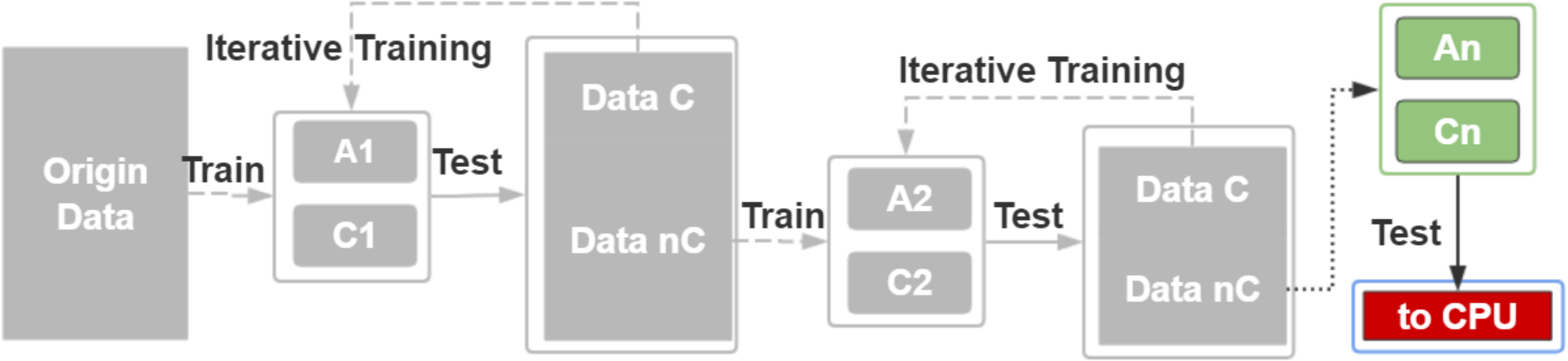
- **Training Process**
  - The original input samples are used to train classifier C1 and approximator A1.
  - Feed the remaining input samples not yet to be recognized by C1 (Data nC) to classifier C2 and approximator A2.



Training Process

Multiple Cascaded Classifiers and Approximators (MCCA)

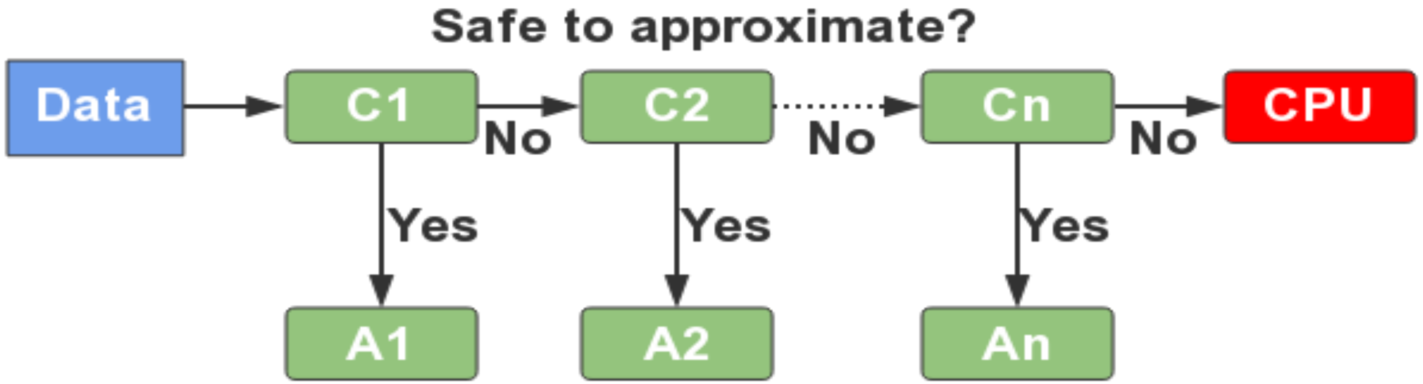
- **Training Process**
  - The original input samples are used to train classifier C1 and approximator A1.
  - Feed the remaining input samples not yet to be recognized by C1 (Data nC) to classifier C2 and approximator A2.
  - Repeat until a specific pair of Cn and An cannot converge.



Training Process

Multiple Cascaded Classifiers and Approximators (MCCA)

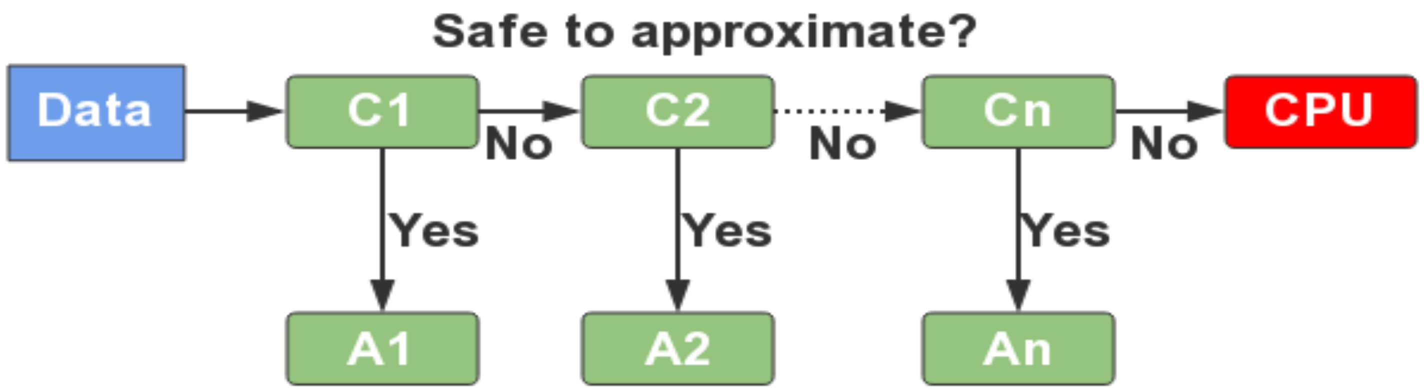
- **Inference Process**
  - If C1 approves, the input data are sent to A1.



Inference Process

Multiple Cascaded Classifiers and Approximators (MCCA)

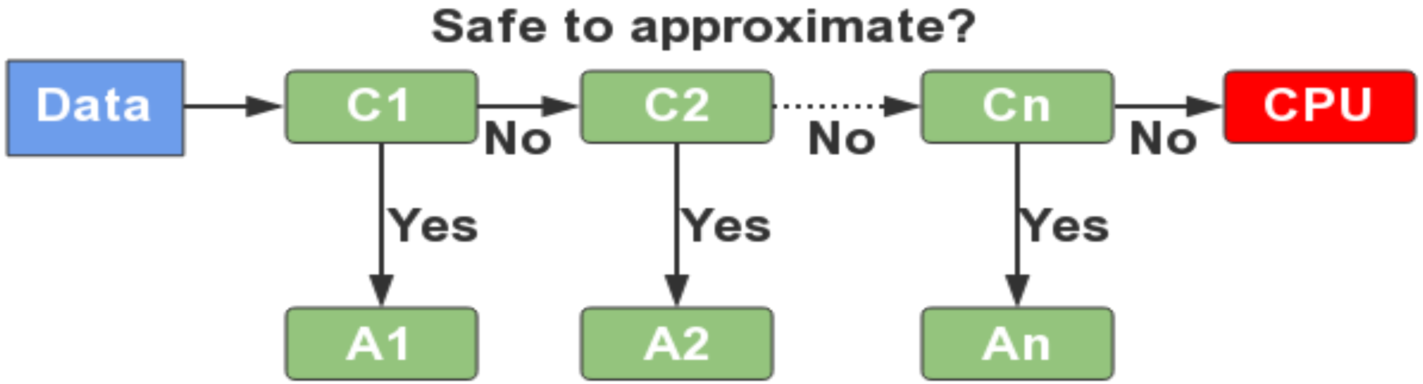
- **Inference Process**
  - If C1 approves, the input data are sent to A1.
  - If C1 disapproves, the input data are sent to the next classifier C2.



Inference Process

Multiple Cascaded Classifiers and Approximators (MCCA)

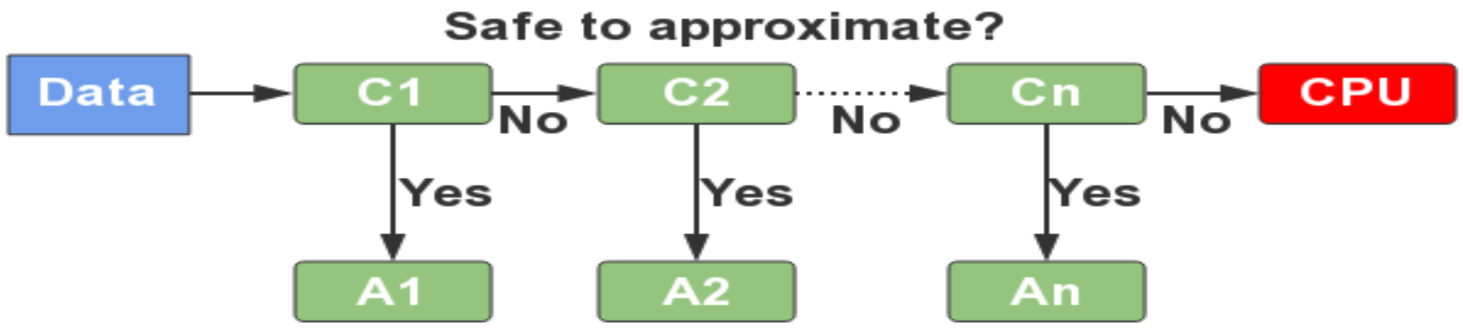
- **Inference Process**
  - If C1 approves, the input data are sent to A1.
  - If C1 disapproves, the input data are sent to the next classifier C2.
  - Repeat until Cn approves.



Inference Process

Multiple Cascaded Classifiers and Approximators (MCCA)

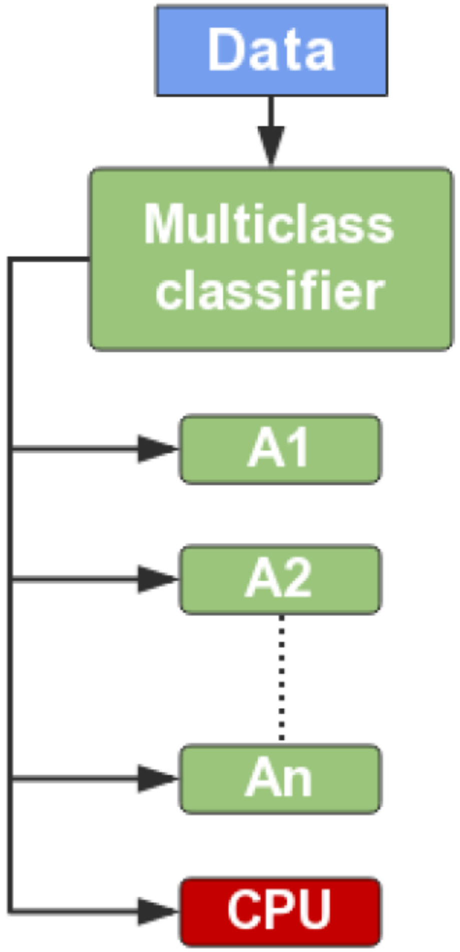
- Inference Process
  - If C1 approves, the input data are sent to A1.
  - If C1 disapproves, the input data are sent to the next classifier C2.
  - Repeat until Cn approves.
- Demerit
  - The time spending on inference is too long



Inference Process

Multiclass-classifier and Multiple Approximators (MCMA)

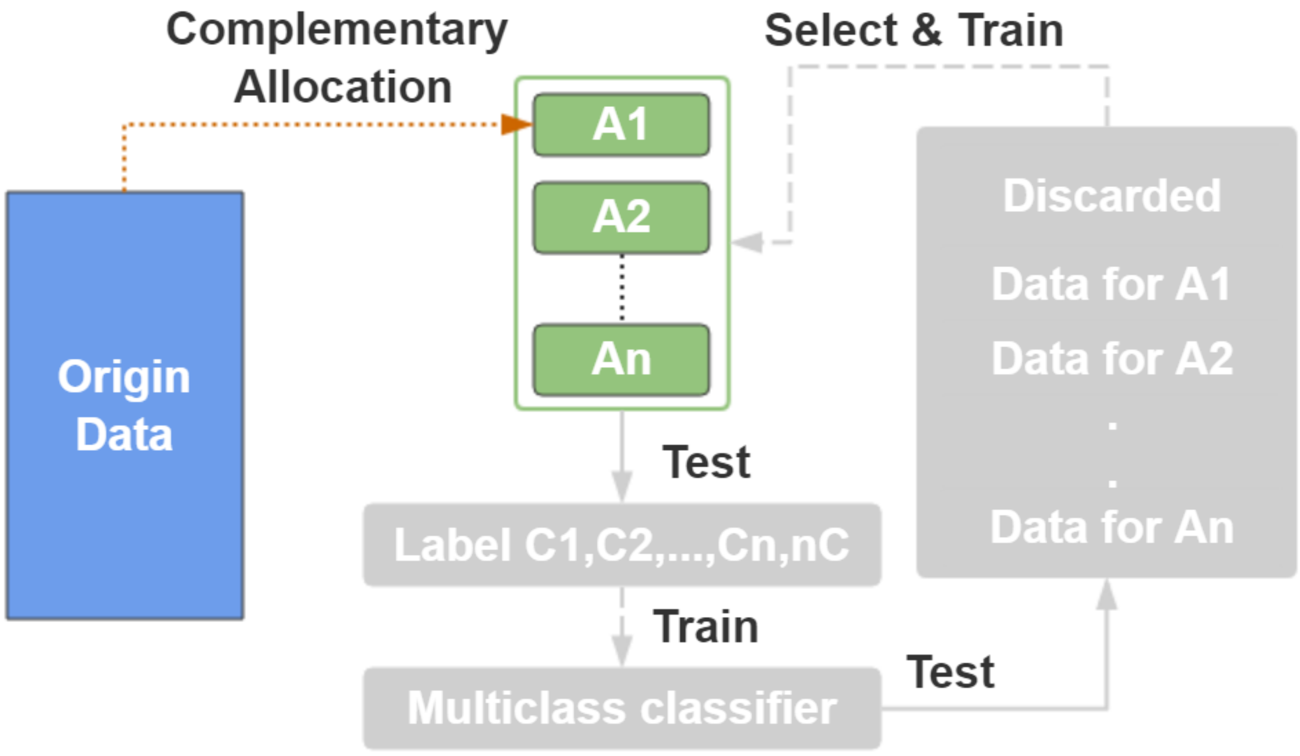
- **Inference Process**
  - The multiclass-classifier predicts which approximator can approximate the input data.



Inference Process

Multiclass-classifier and Multiple Approximators (MCMA)

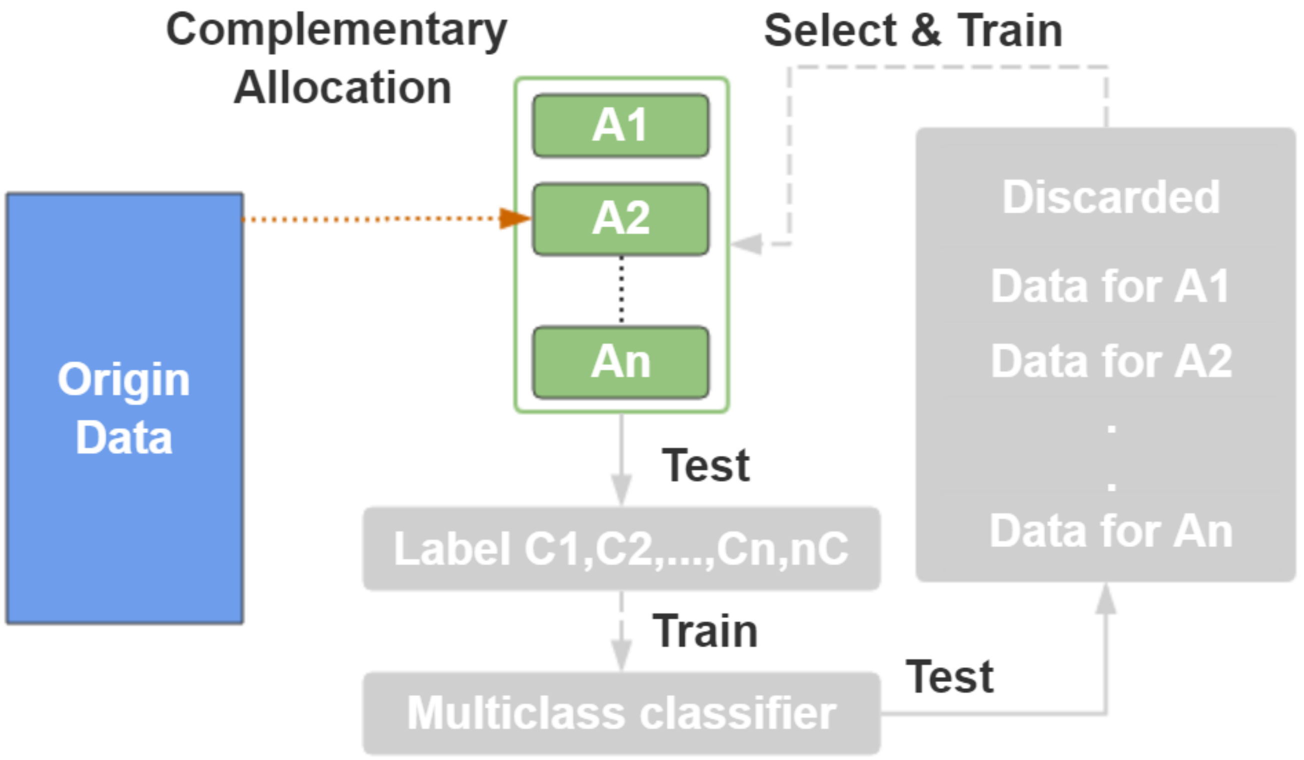
- **Complementary training**
  - Test A1 with all data, produce the the label C1 for any input sample that A1 can safely approximate



Training Process

Multiclass-classifier and Multiple Approximators (MCMA)

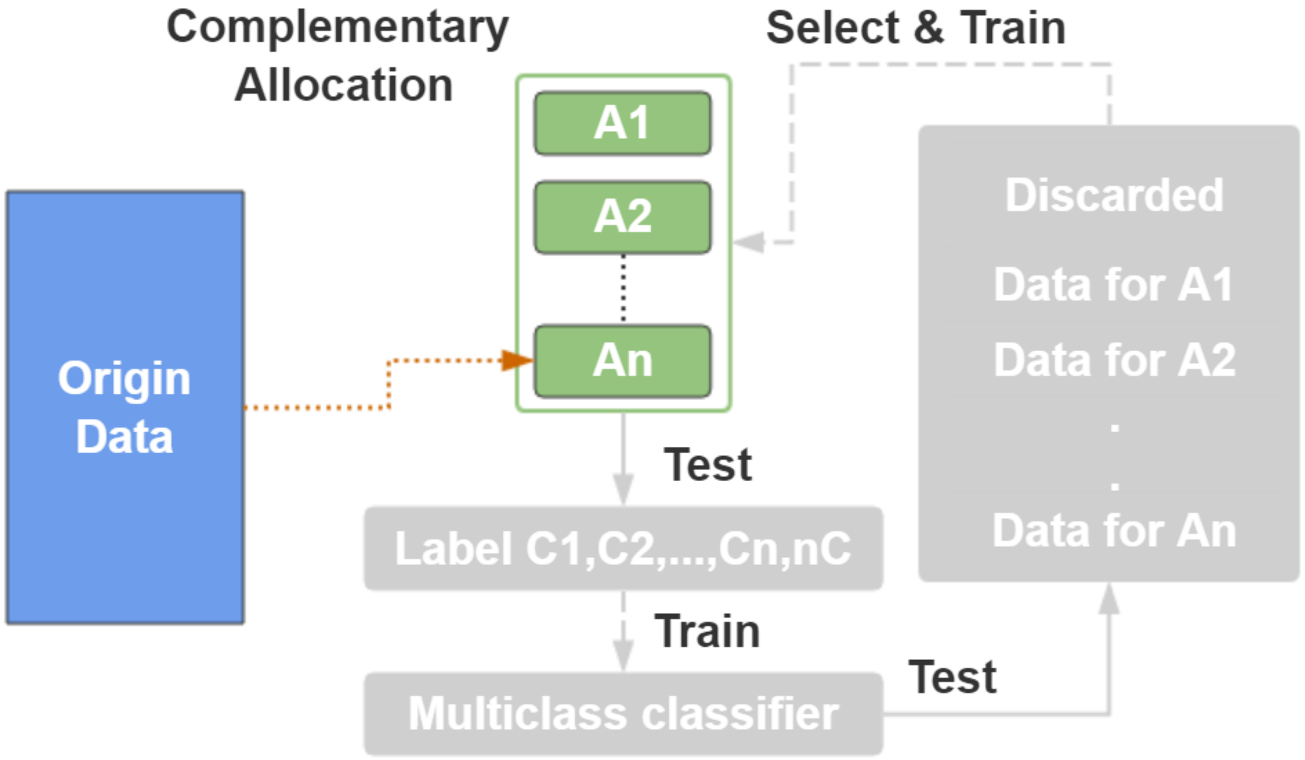
- **Complementary training**
  - Test A1 with all data, produce the the label C1 for any input sample that A1 can safely approximate
  - Test A2 with the remaining data, produce the the label C2 for any input sample that A2 can safely approximate



Training Process

Multiclass-classifier and Multiple Approximators (MCMA)

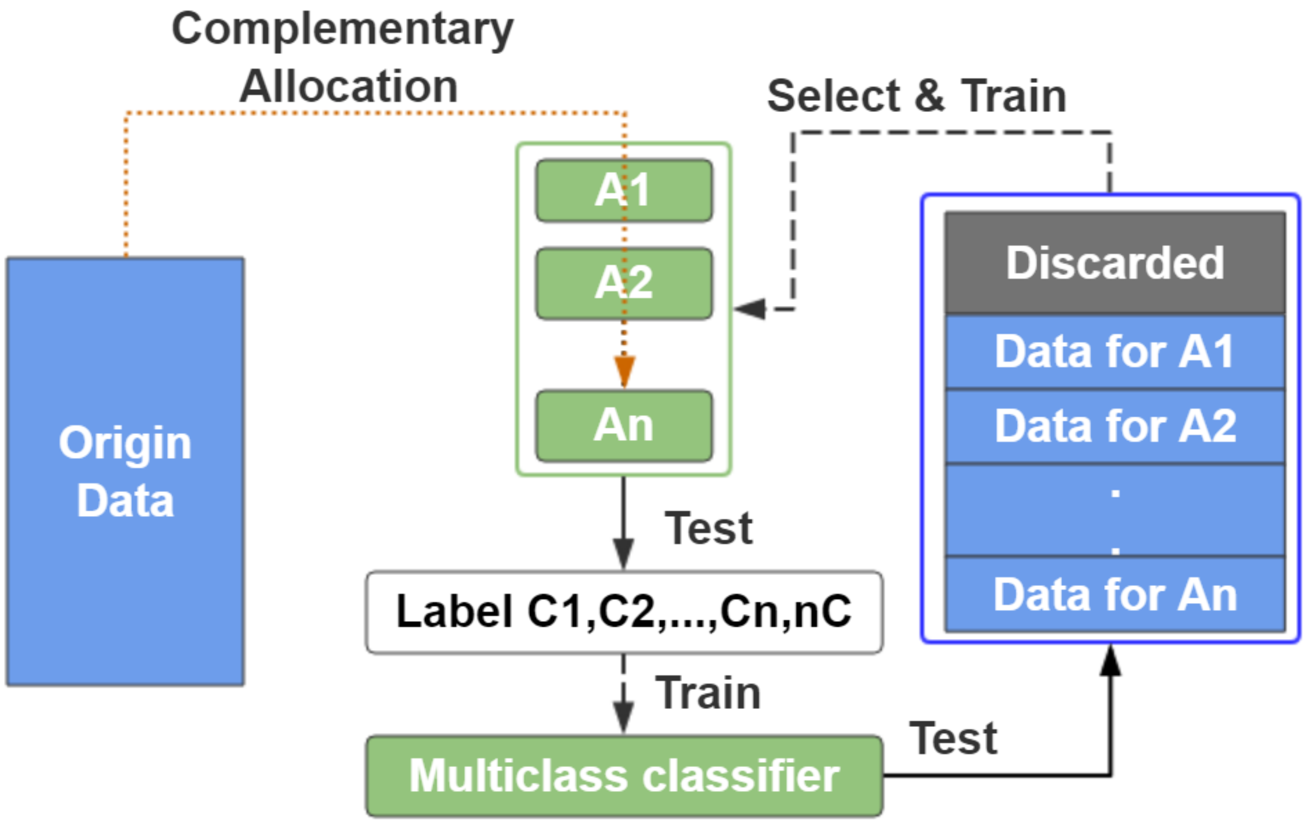
- **Complementary training**
  - Test A1 with all data, produce the the label C1 for any input sample that A1 can safely approximate
  - Test A2 with the remaining data, produce the the label C2 for any input sample that A2 can safely approximate
  - Repeat until test An, the remaining input samples without any label are labeled as nC.



Training Process

Multiclass-classifier and Multiple Approximators (MCMA)

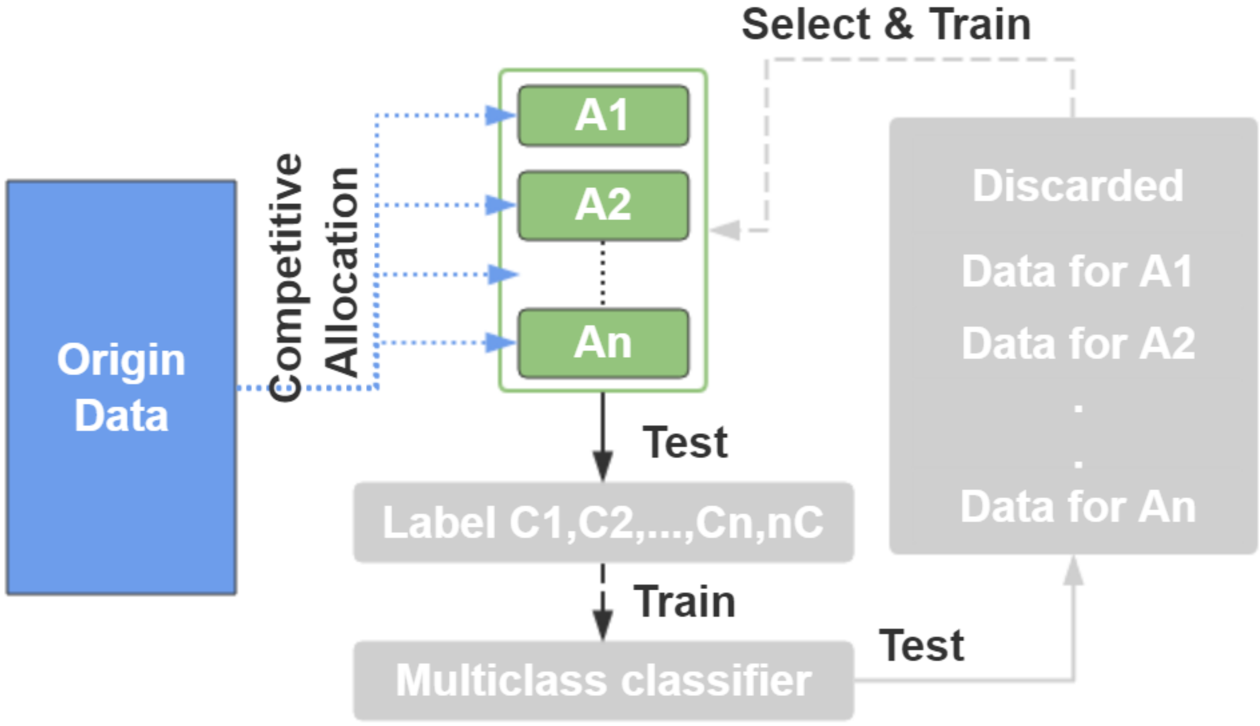
- **Complementary training**
  - Test A1 with all data, produce the the label C1 for any input sample that A1 can safely approximate
  - Test A2 with the remaining data, produce the the label C2 for any input sample that A2 can safely approximate
  - Repeat until test An, the remaining input samples without any label are labeled as nC.
  - Train the multiclass-classifier and approximators using iterative training.



Training Process

Multiclass-classifier and Multiple Approximators (MCMA)

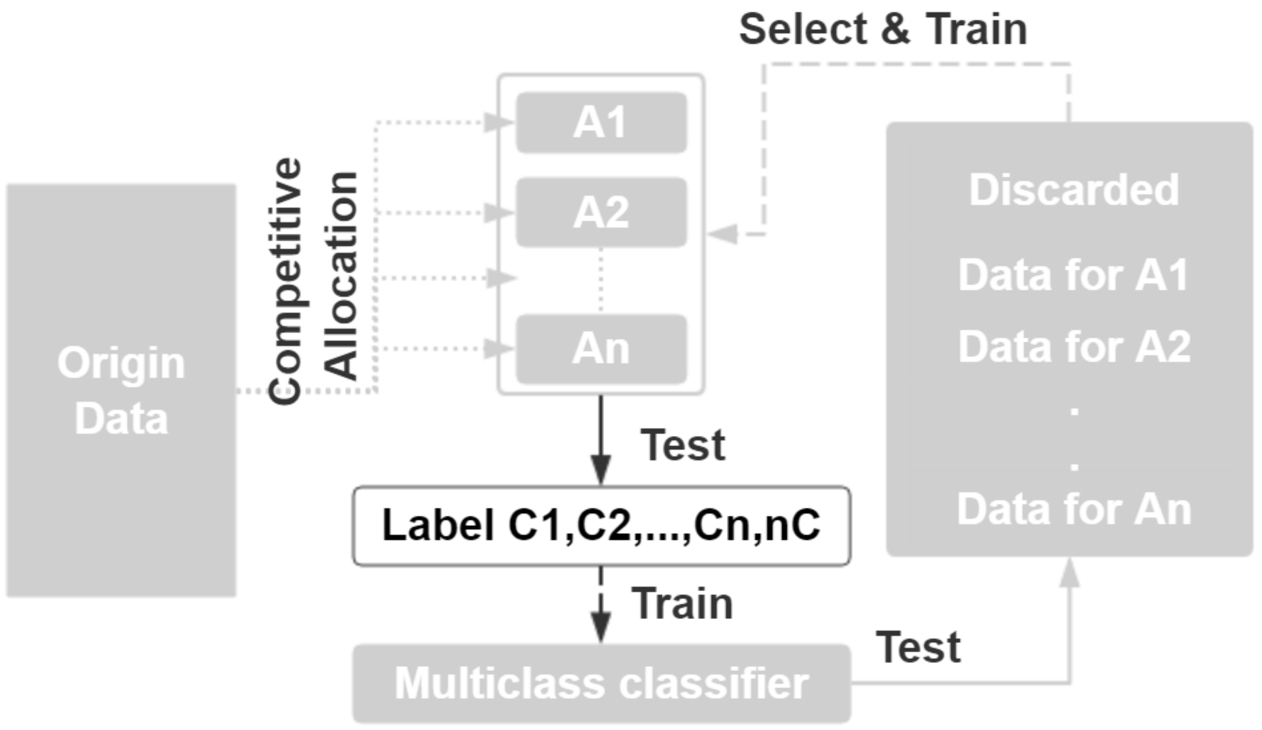
- **Competitive training**
  - Test A1 with all data, obtain the approximation error.
  - Test A2 with all data, obtain the approximation error.
  - …
  - Test An with all data, obtain the approximation error.



Training Process

Multiclass-classifier and Multiple Approximators (MCMA)

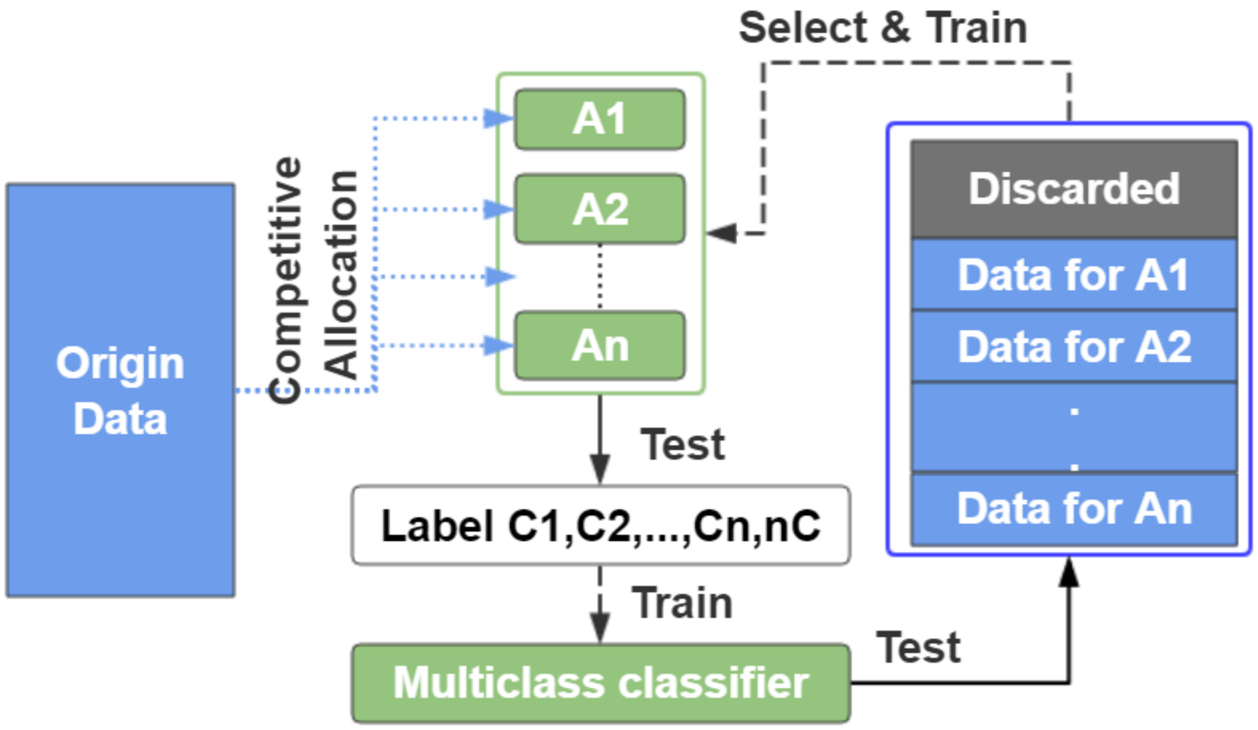
- **Competitive training**
  - Test A1 with all data, obtain the approximation error.
  - Test A2 with all data, obtain the approximation error.
  - …
  - Test An with all data, obtain the approximation error.
  - Generate the label for each data according to the lowest approximation error.



Training Process

Multiclass-classifier and Multiple Approximators (MCMA)

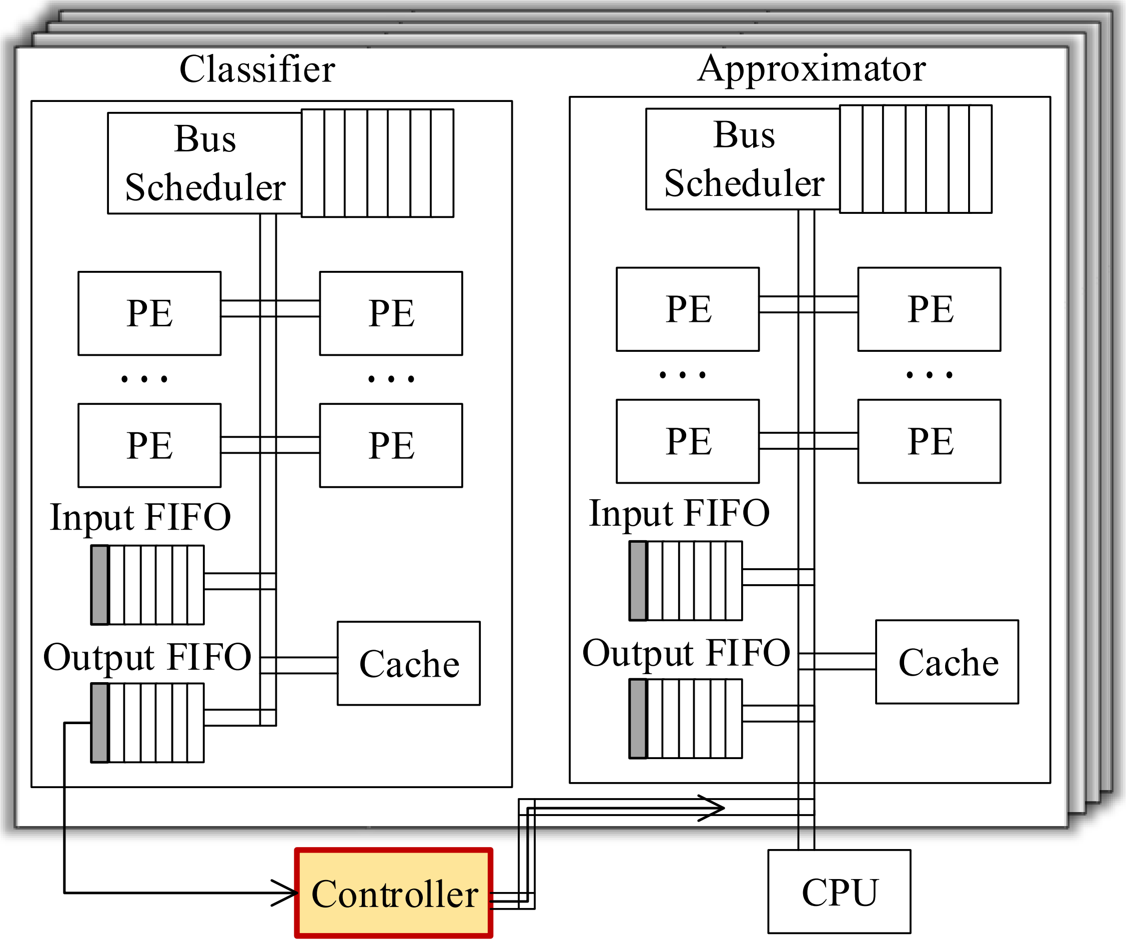
- **Competitive training**
  - Test A1 with all data, obtain the approximation error.
  - Test A2 with all data, obtain the approximation error.
  - …
  - Test An with all data, obtain the approximation error.
  - Generate the label for each data according to the lowest approximation error.
  - Train the multiclass-classifier and approximators using iterative training.



Training Process

Hardware design

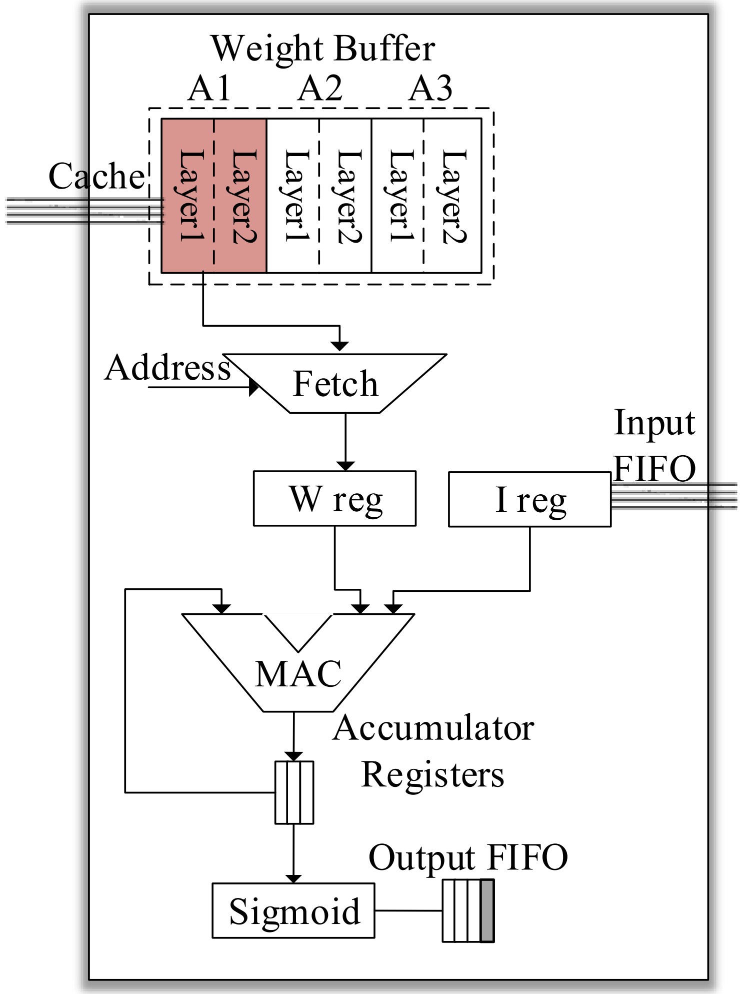
- Add a Controller to control the weight buffer inside the PE.



The overall NPU design

Hardware design

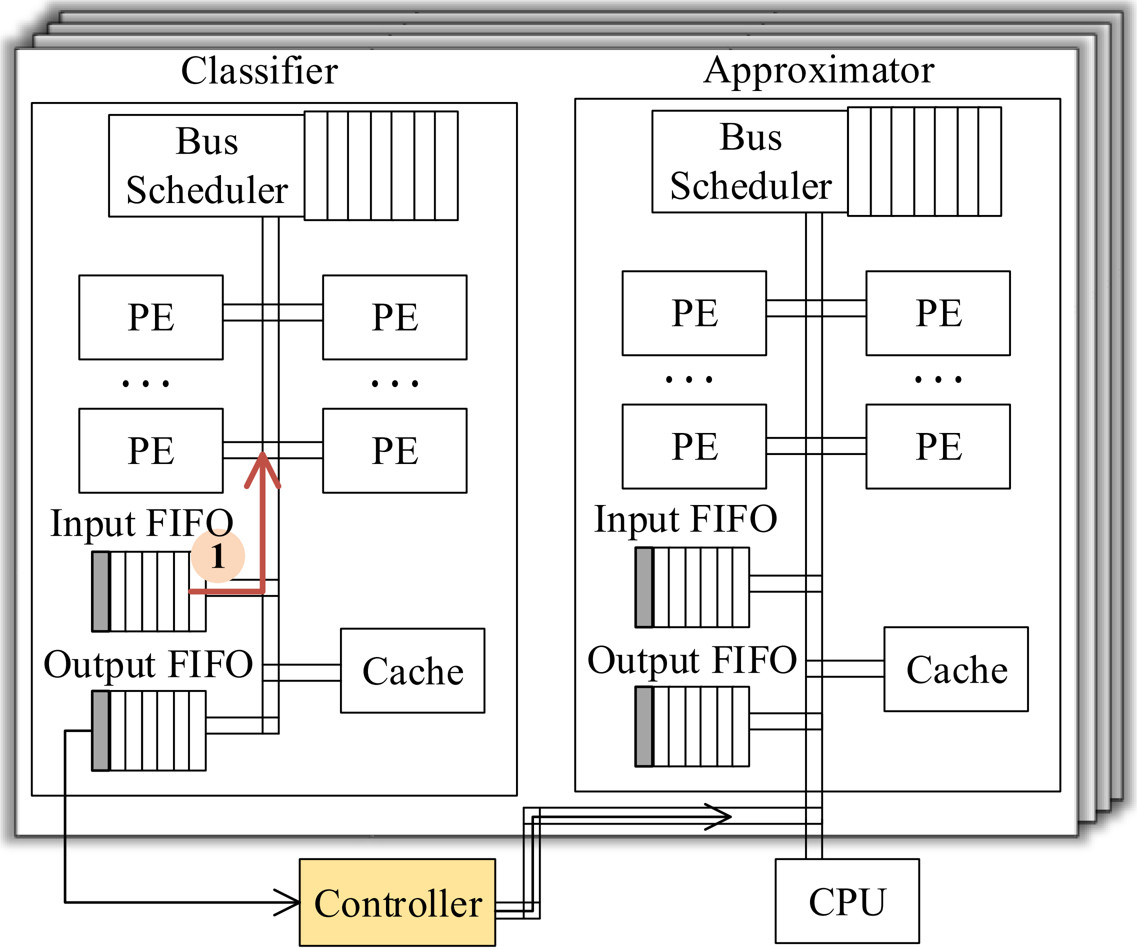
- Add Controller to control multiple approximators.
- Weight buffer receives the signal from the controller, and then sechedule approximators.

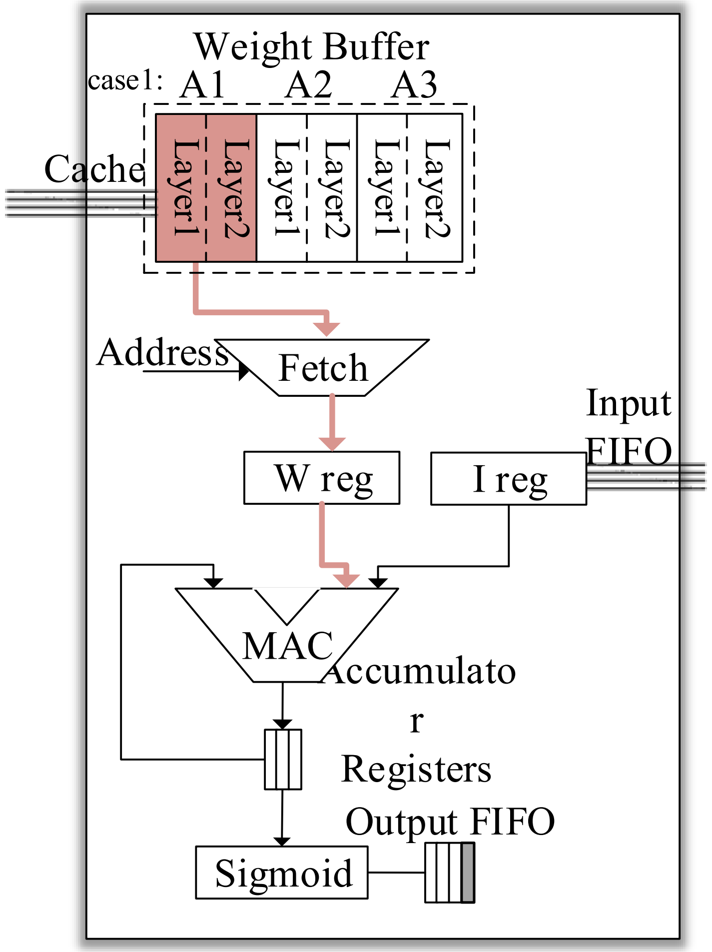


The detail PE design

Hardware design

- Read data from Input FIFO.

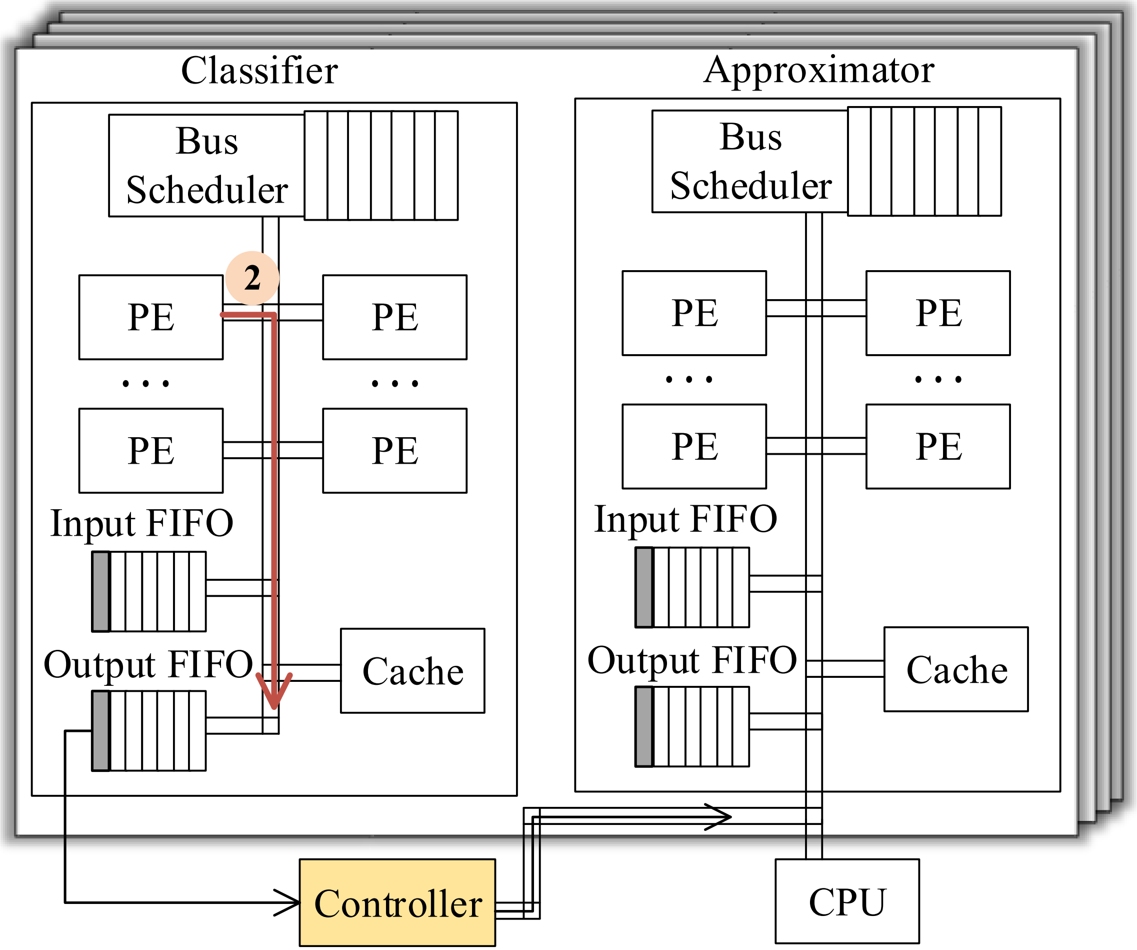


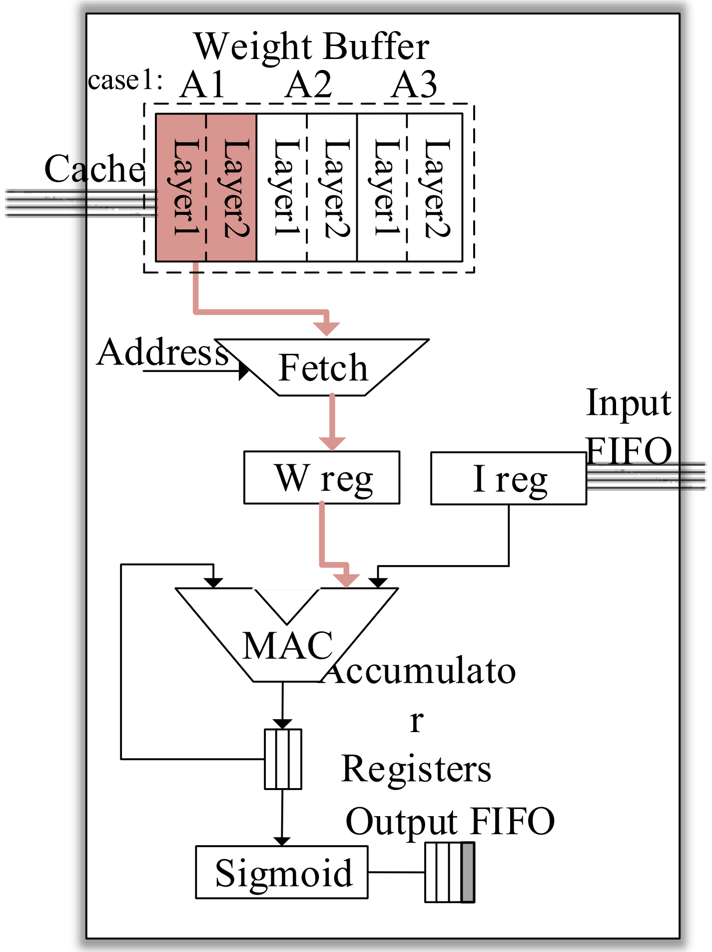


Data flow

Hardware design

- Conduct vector multiplication in PE.

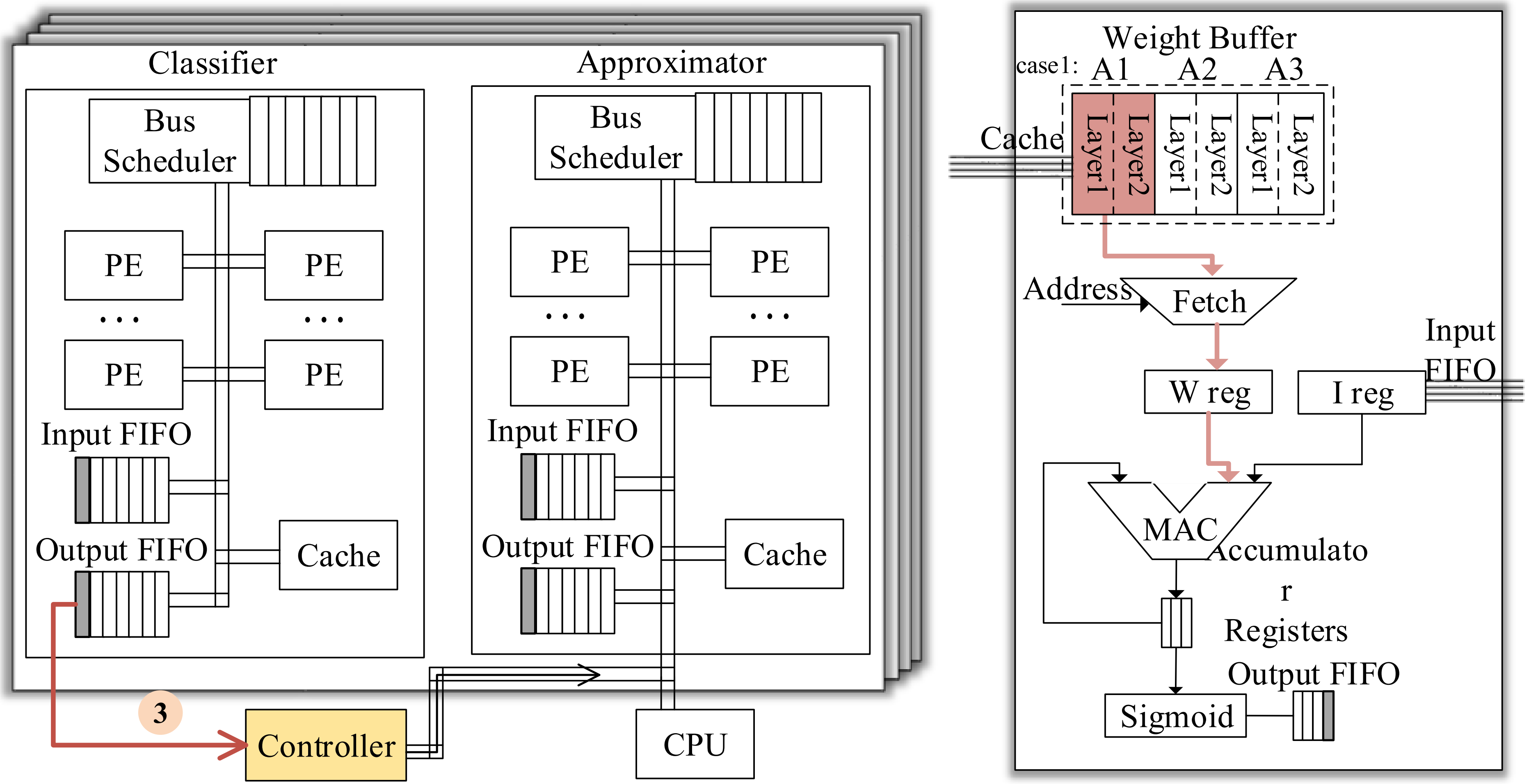




Data flow

Hardware design

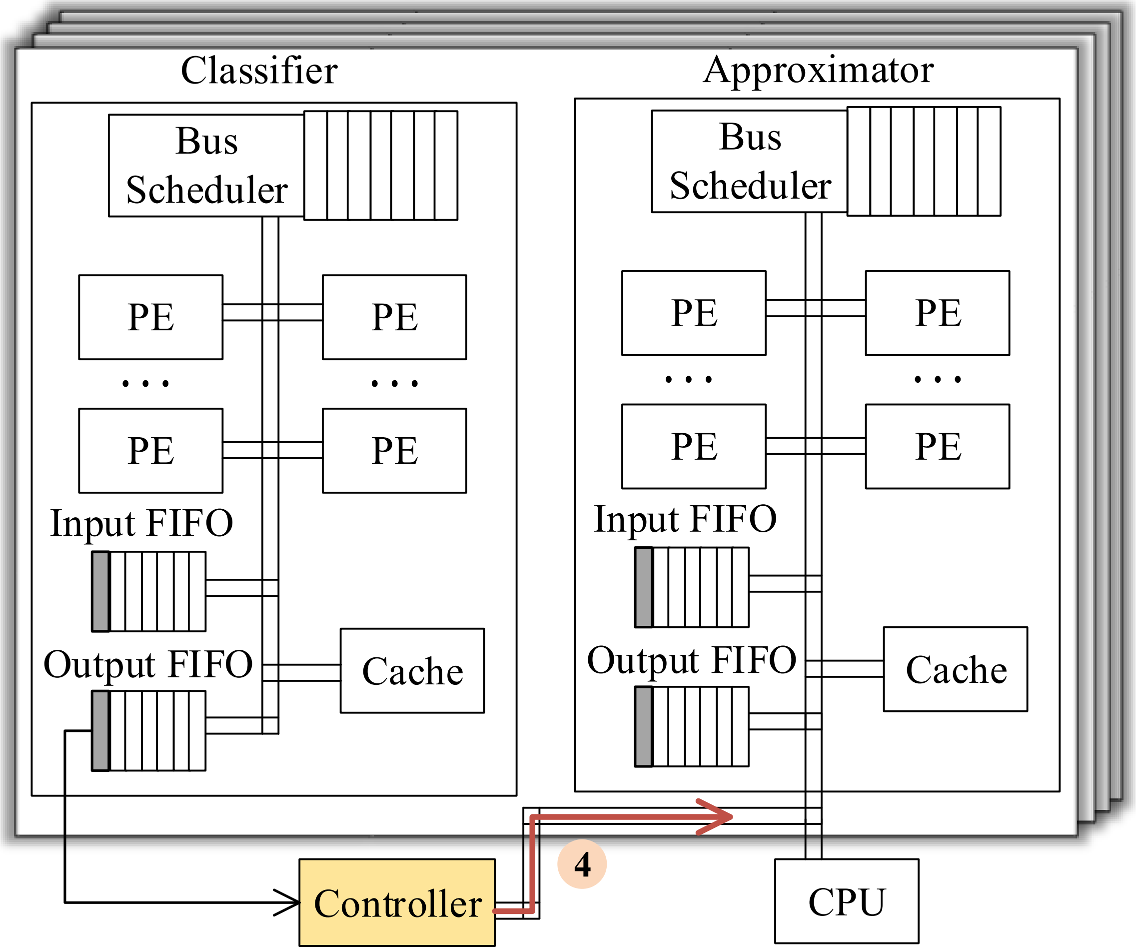
- Controller send signal to CPU or Approximator.

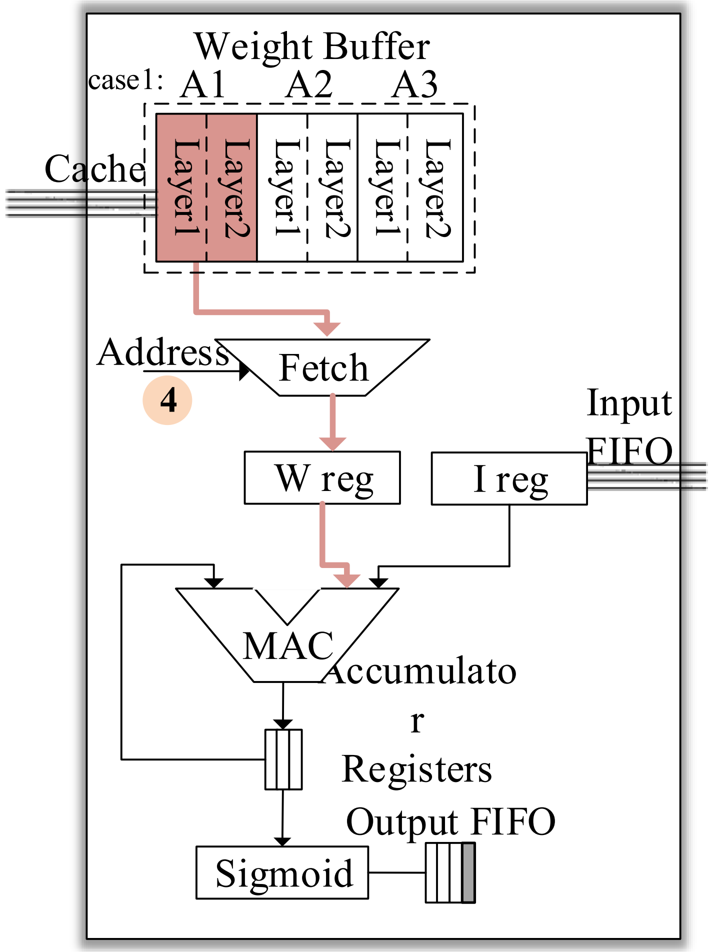


Data flow

Hardware design

- If approximator invoked, fetch corresponding approximator's weight.

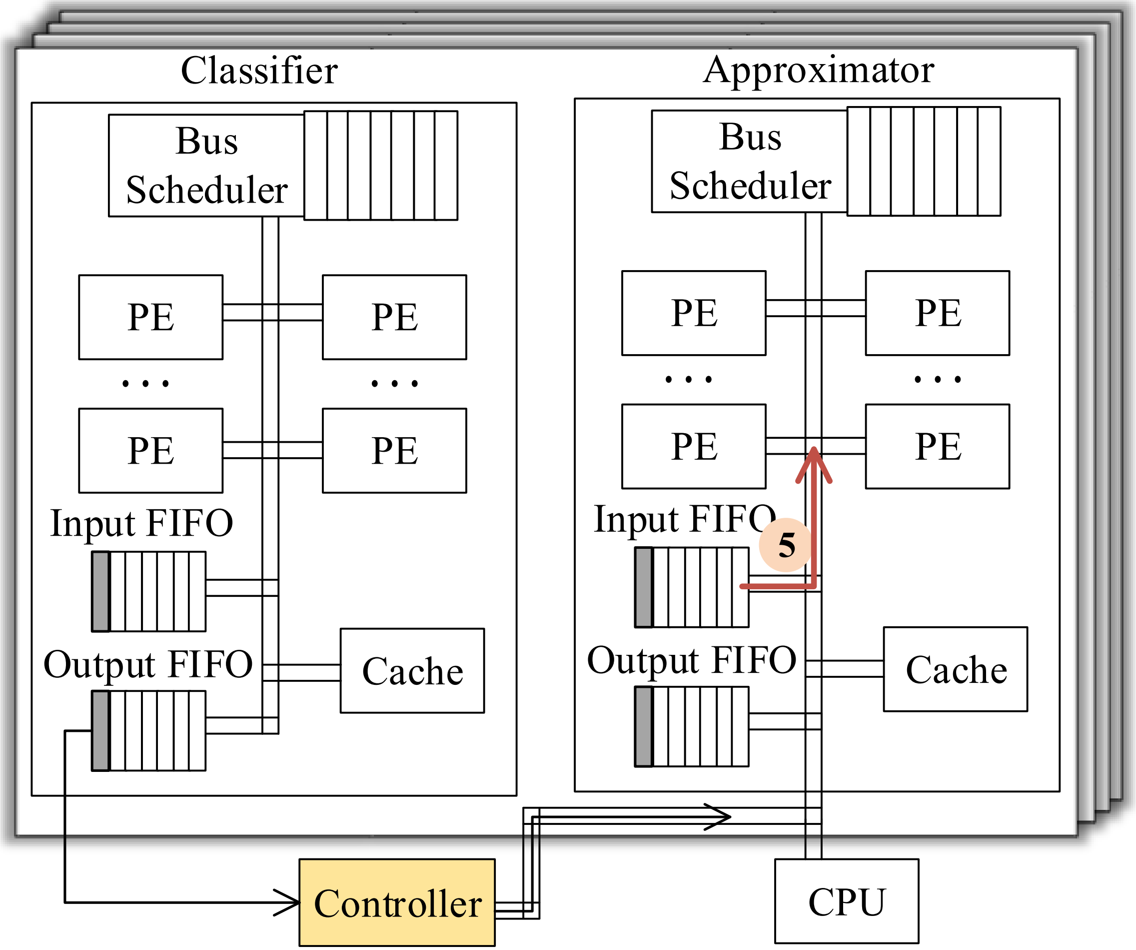


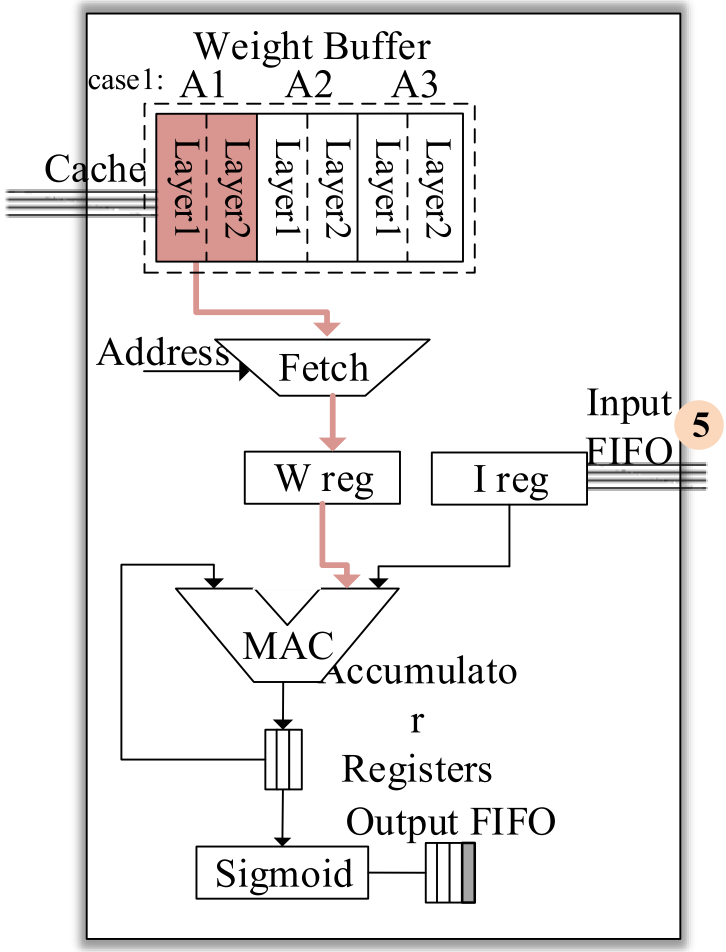


Data flow

Hardware design

- Conduct vector multiplication in PE.

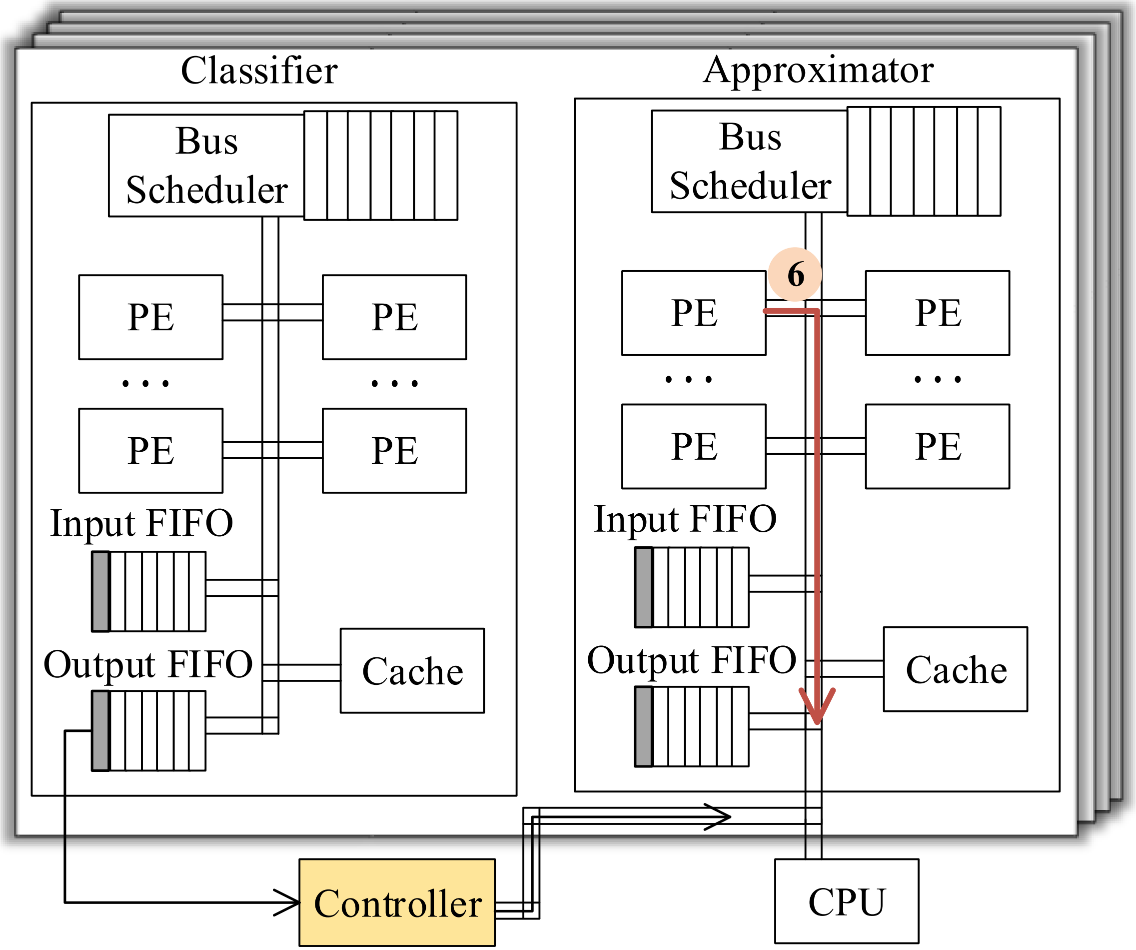


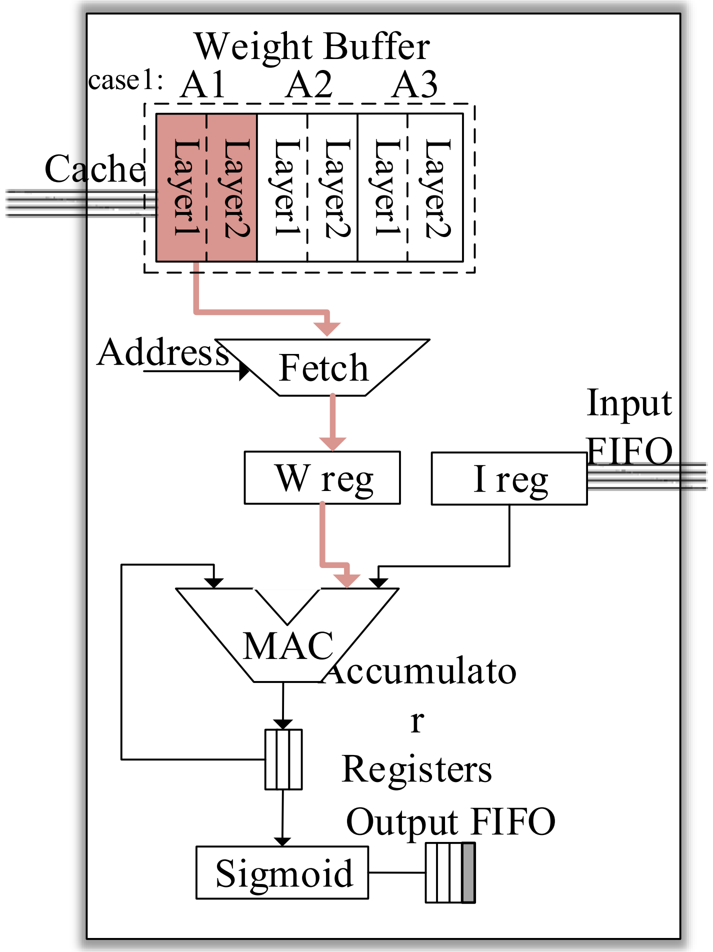


Data flow

Hardware design

- Send back the result from PE to output FIFO.

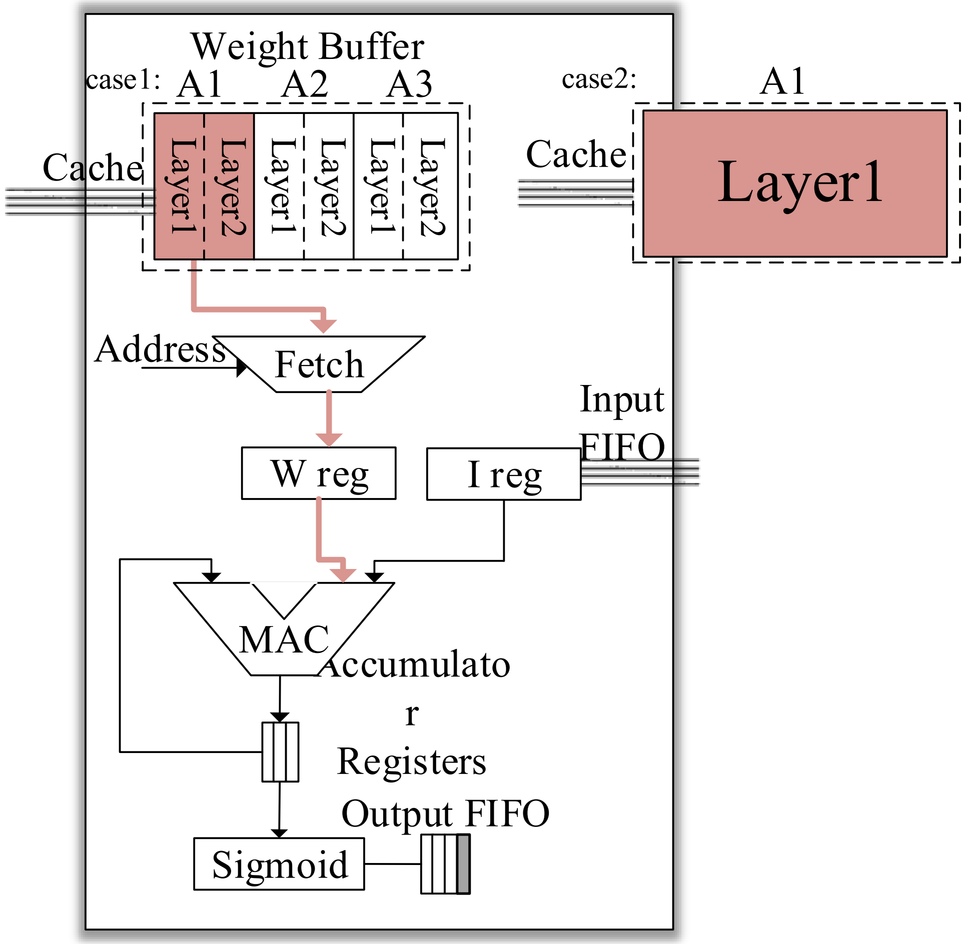




Data flow

Hardware design

- Load the weights layer by layer.



The detail PE design

1 Background

2 Related works and Motivation

3 Proposed Method

4 Experiment Results

5 Conclusion

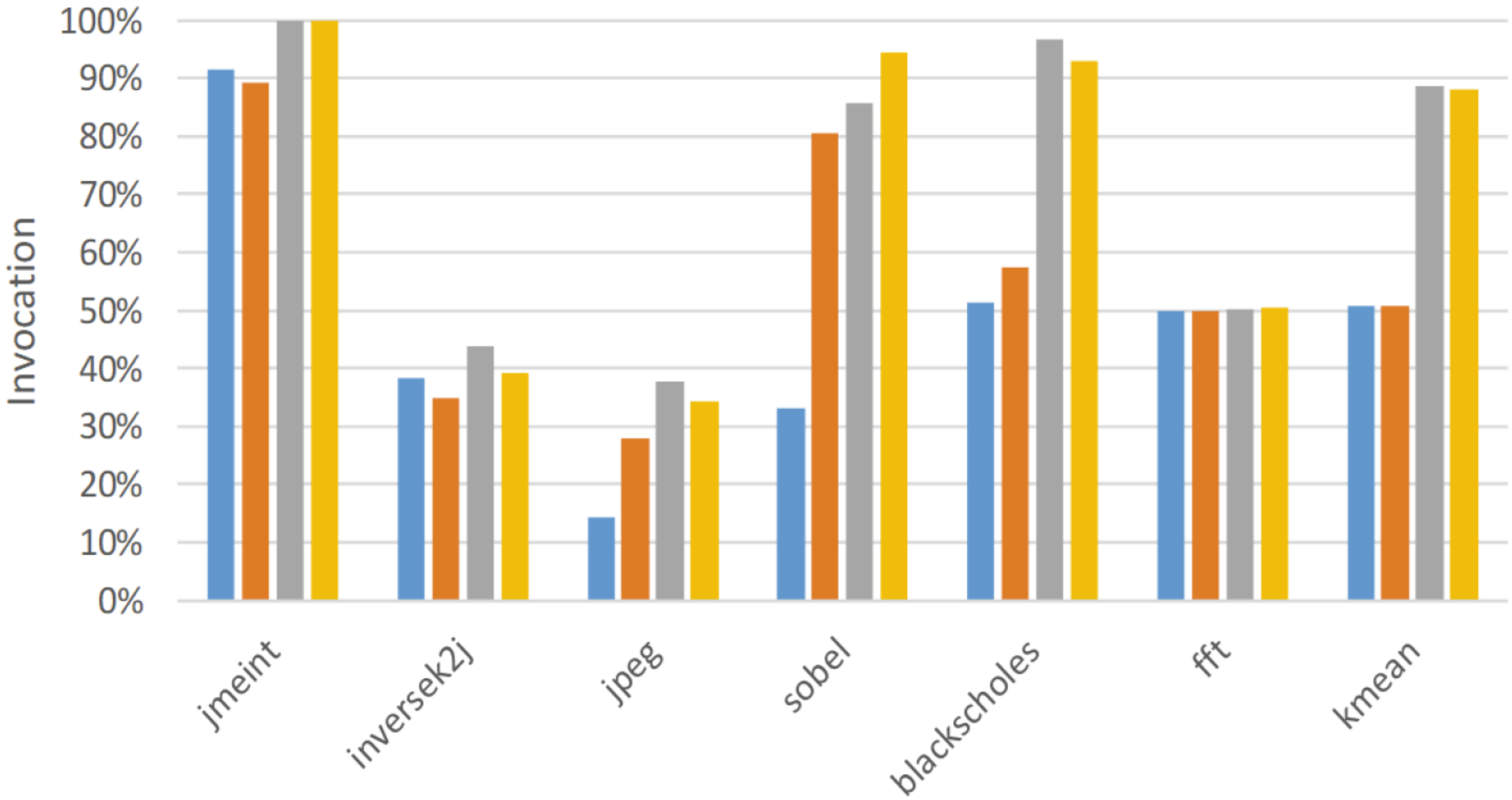
Experimental setup

- Compared with One-pass[ISCA'16] and Iterative training[DAC'17]
- 8 benchmark applications

#	Benchmark	Domain	Train Data	Test Data	Approximator Topology	Classifier Topology
1	Black-Scholes	Financial Analysis	70K options	30K options	6->8->1	6->8->2(4)
2	FFT	Signal Processing	8K fp numbers	3K fp numbers	1->2->2->2	1->2->2(4)
3	Inversek2j	Robotics	70K (x,y) pairs	30K (x,y) pairs	2->8->2	2->8->2(4)
4	Jmeint	3D gaming	70K traingles	30K traingles	18->32->16->2	18->16->2(4)
5	JPEG encoder	Compression	512*512 pixel color image	512*512 pixel color image	64->16->64	64->16->2(4)
6	K-means	Machine Learning	100K pairs of (r,g,b) points	50K pairs of (r,g,b) points	6->8->4->1	6->8->4->2(4)
7	Sobel	Image Processing	512*512 pixel color image	512*512 pixel color image	9->8->1	9->8->2(4)
8	Bessel	Scientific Computing	70K fp pairs	30K fp pairs	2->4->4->1	2->4->2(4)

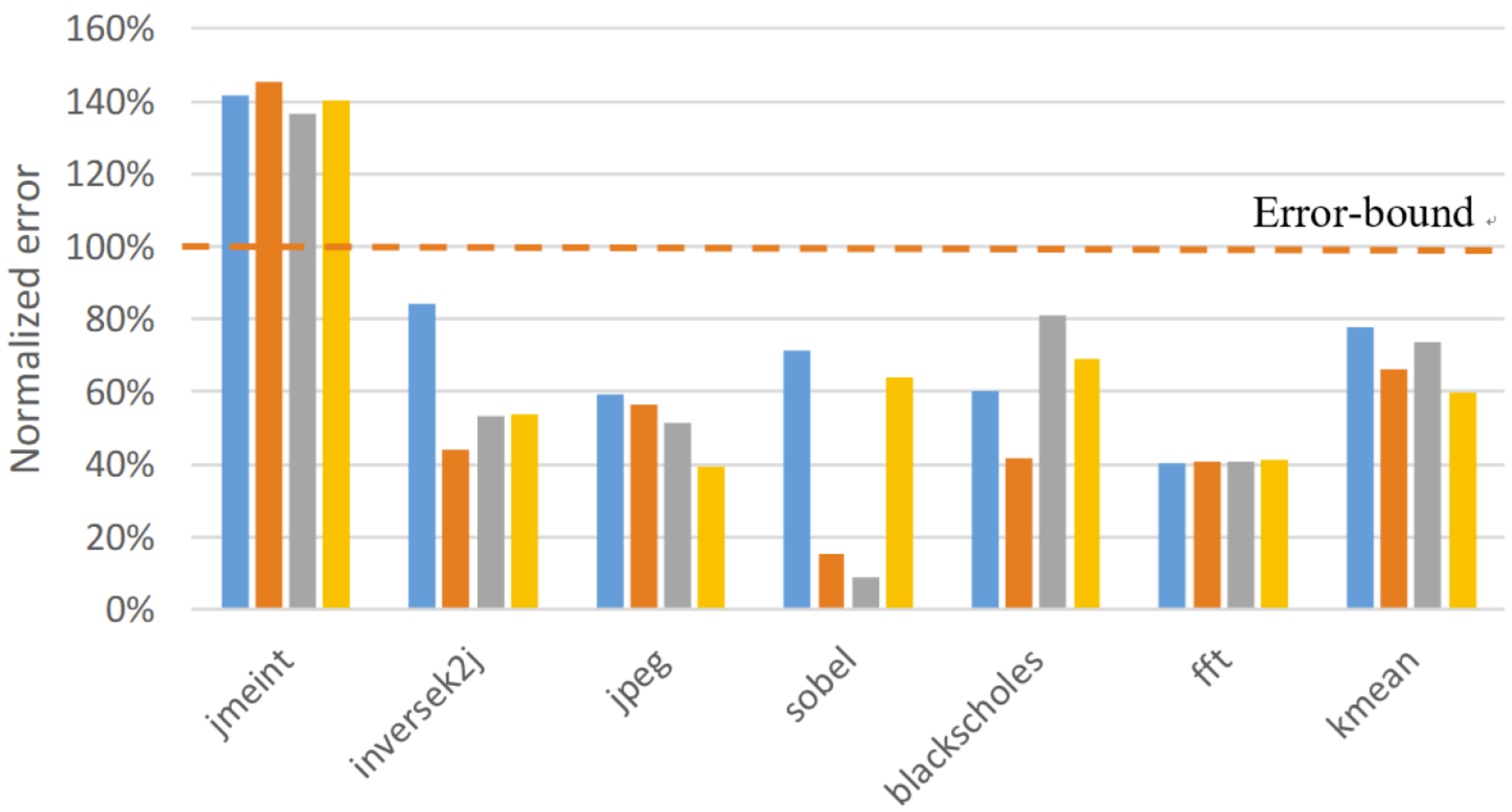
Experiment Results

- Invocation increase **20%~30%** on average.
- Invocation increase **40%+** in sobel or kmeans benchmark.



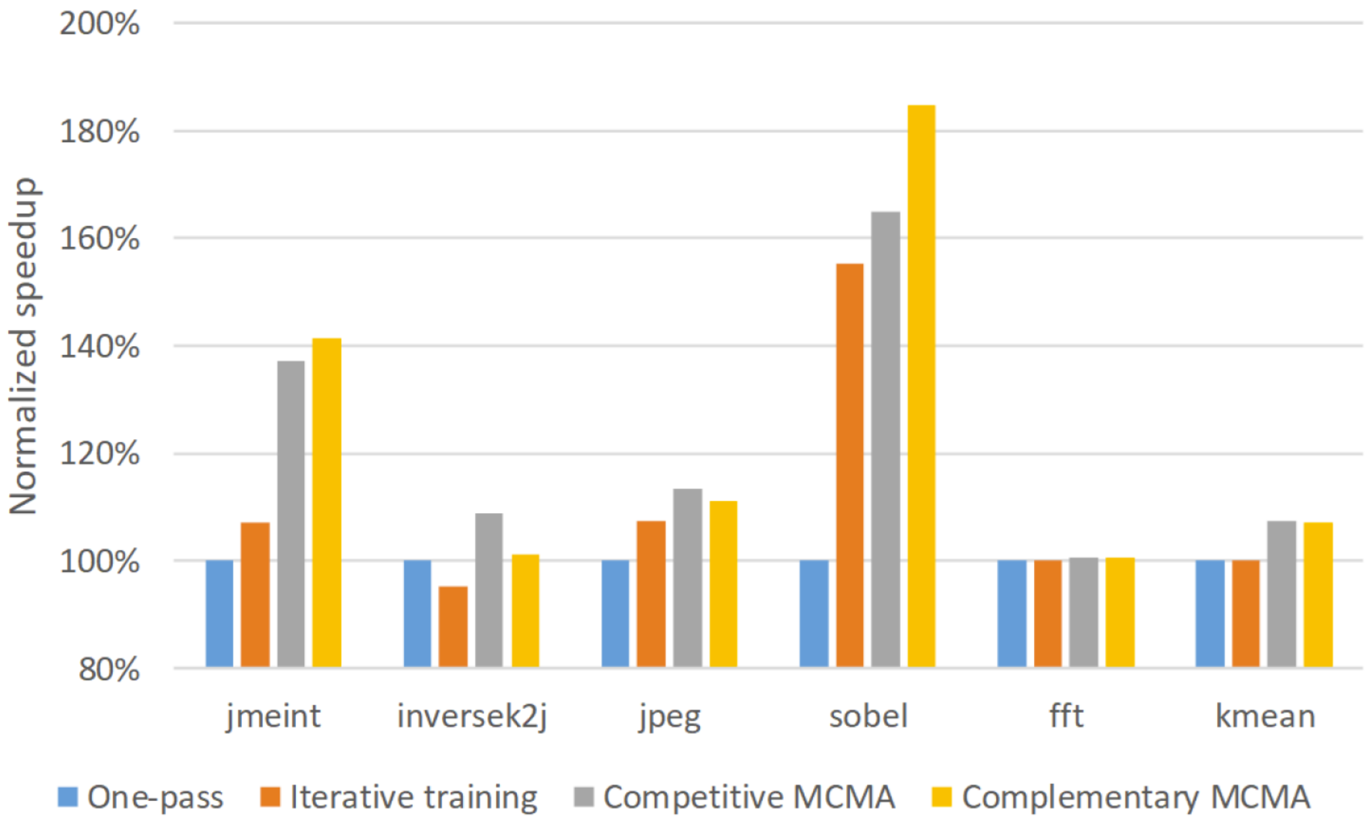
Experiment Results

- The approximation error is **below** the error bound in **most** benchmarks.



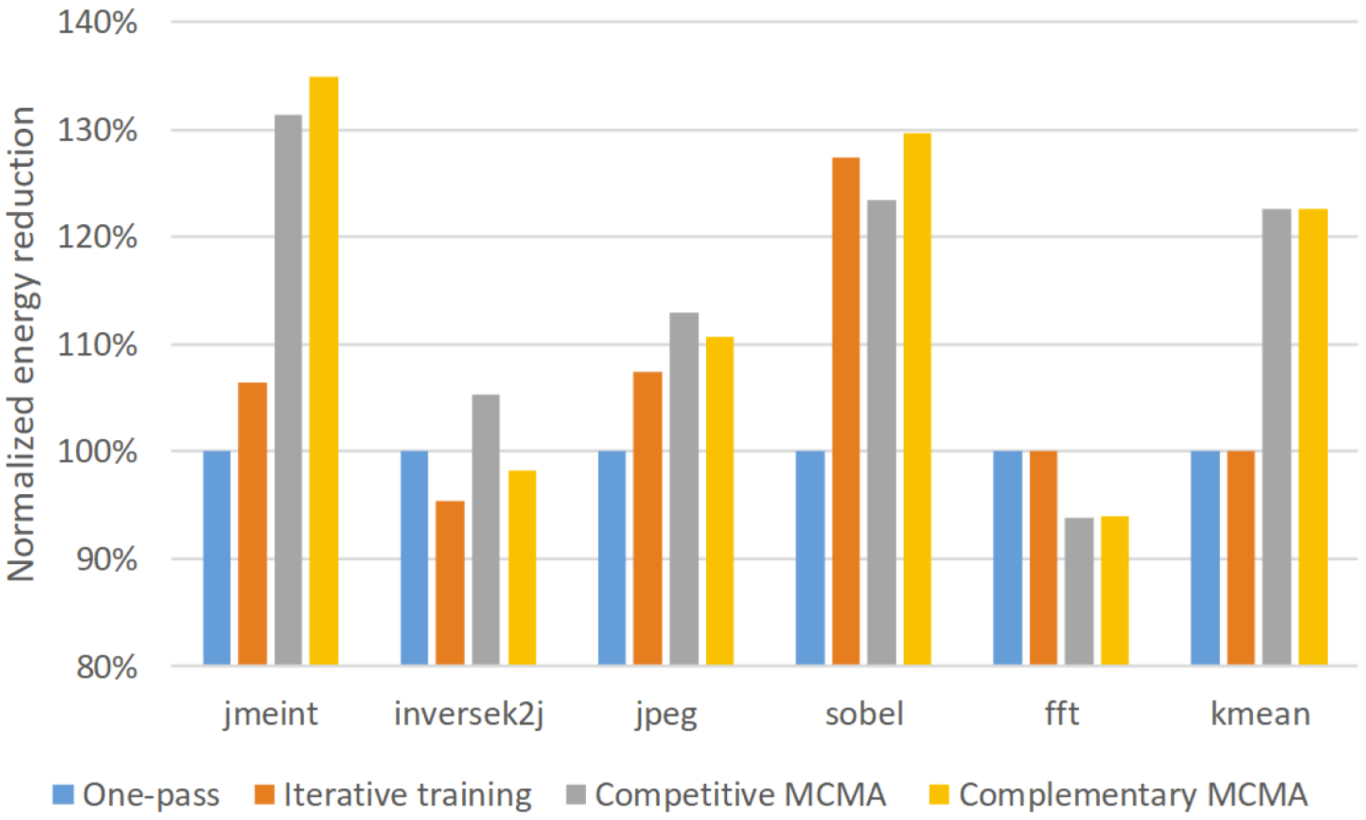
Experiment Results

- The average speedup is **1.23x** compared with one-pass method.



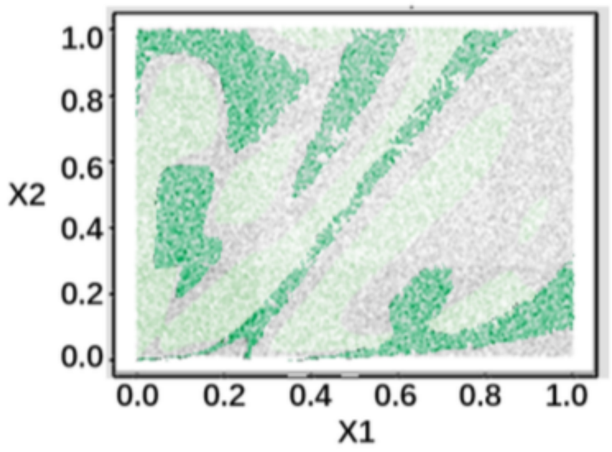
Experiment Results

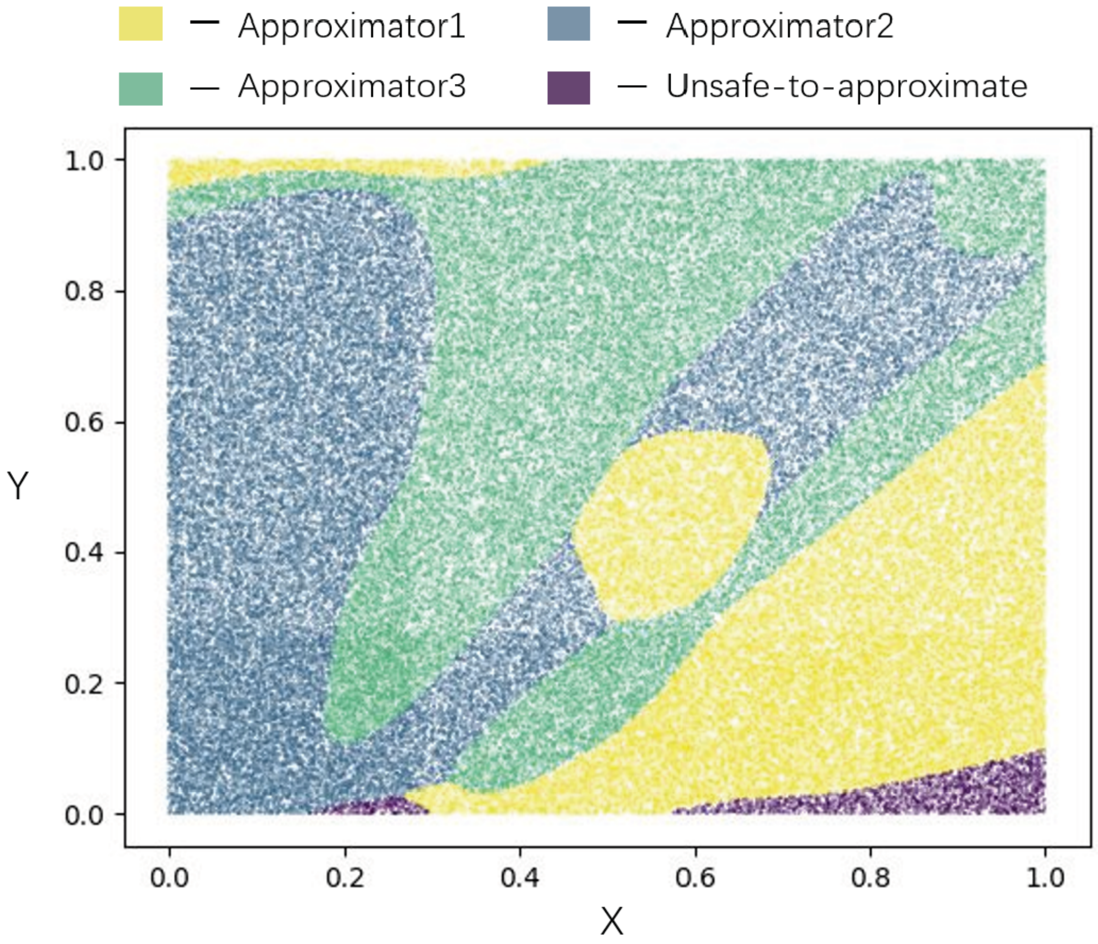
- The average energy reduction is **1.15x** compared with one-pass method.



Experiment

- Almost all samples have a corresponding approximator that can approximate it





1 Background

2 Related works and Motivation

3 Proposed Method

4 Experiment Results

5 Conclusion

Thanks for listening!

Invocation-driven Neural Approximate Computing with a Multiclass-Classifier and Multiple Approximators

Zhuoran Song (宋卓然)
Professor Li Jiang (蒋力)
Advanced Computer Architecture Laboratory
Shanghai Jiao Tong University

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY