A FPGA friendly approximate computing framework with hybrid Neural networks

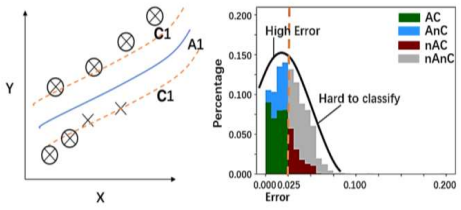
Haiyue Song, Xiang Song, Chengwen Xu, Hao Dong, Tianjian Li and Li Jiang

Department of CS, Shanghai Jiao Tong University

Approximation with approximator and classifier

Neural approximate computing is promising to gain energy-efficiency and performance by the tradeoff of tolerable errors.

One way is to use classifier-approximator hybrid architecture where classifier tells the approximator which part is safe-to-approximate and discard the unsafe part.

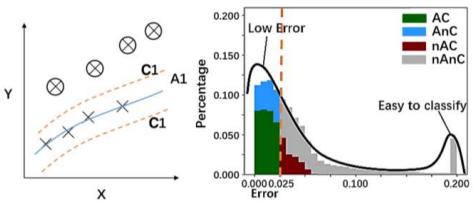


(a) One-pass training method.

Using Iterative method

And when we train the classifier-approximator hybrid architecture iteratively the performance will be improved.

What about the discarded data?

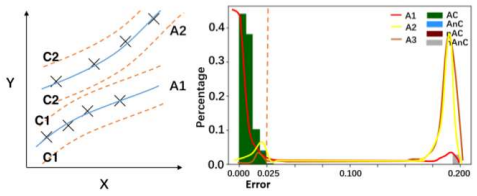


(b) Iterative training method

Using multiple approximators

We can use another approximator to approximate them!

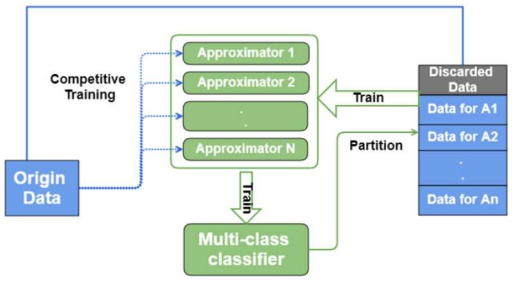
We proposes a multi-class classifier and multiple approximators (MCMA) architecture. The idea is using more than one approximators so that each one can approximate smaller part of data but more precisely.



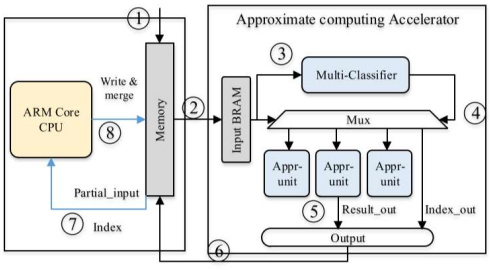
(c) Proposed MCMA Architecture and its training method.

Model

In the MAMC, firstly the origin data is used to train all the approximators. Then the approximators tells the classifier whether a data is suitable for it. After that the classifier partition the data and each approximator is feed by only a small part of data by which the precise will be improved.



The iterative training process for MAMC



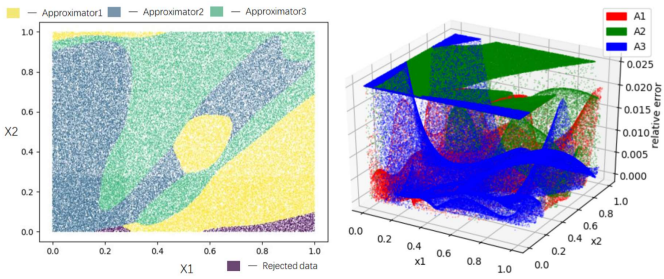
Data path for the ACA architecture

Why MAMC, not a bigger classifier and approximator?

Because only one Appr-unit will be activated which will be energy efficient!

Result

Each approximator in MCMA can have its own specialty in fitting a specific cluster of samples. And each approximator may output results with large error in some area. However, with the cooperation of the three approximators and the multi-class classifier, the MCMA architecture is able to approximate a large portion of data under the error bound.



Data samples of Bessel distributed in MCMA (left) and their relative error distribution

Invocation between original and MCMA methods