Neural Acceleration for General-Purpose Approximate Programs

Approximate computing research: many applications can tolerate small errors during execution.

Technology trends mean that future chips will benefit from specialization and acceleration.

Hardware neural networks are:
- flexible
- low power
- parallel
- regular
- fault tolerant
- parallel

We show that neural networks can approximate many functions written in conventional programming languages. We propose an algorithmic transformation and hardware accelerator that improves programs’ performance and energy efficiency with very little accuracy loss.

The programmer marks code that is hot, approximable, and has well-defined inputs and outputs. Developers also provide a small set of representative test inputs. The rest of the process is automatic.

Code observation:
- training pairs
- test inputs
- instrumentation

Training:
- backpropagation
- topology selection

Code generation:
- NPU scheduling
- insert NPU instructions

Core

6 applications
MARSSx86 simulation
McPAT/CACTI for power

2.3\times mean application speedup
0.8x – 11x

3.0\times mean energy reduction
1.1x – 21x

<10% output quality loss
3.4% – 9.6%

Tight coupling allows low-latency communication with the core.