PULP-NN: Open-Source Library for QNNs Inference on RISC-V Based PULP Cluster

RISC-V Workshop, Zürich

Angelo Garofalo ¹, Manuele Rusci¹, Francesco Conti¹,²,
angelo.garofalo@unibo.it, manuele.rusci@unibo.it, f.conti@unibo.it

Davide Rossi¹, Luca Benini ¹,²,
davide.rossi@unibo.it, luca.benini@unibo.it

¹Department of Electrical, Electronic and Information Engineering
²Integrated Systems Laboratory
Embedded Machine Learning (Deep Learning)

Issue: Trade-off the high computational and memory requirements of ML (DL) with strongly constrained IoT end-nodes

DNN Model, Quantization, pruning & Hardware/Software optimizations

IoT demands for more Embedded Intelligence

- Low latency
- Efficient use of network bandwidth
- Less power
- Reduce Cost
- Privacy
- Reliability

The Internet of Things

An explosion of connected possibility

- Low latency
- Efficient use of network bandwidth
- Less power
- Reduce Cost
- Privacy
- Reliability
HW: IoT-Edge MCUs

- High Performance
- Energy Efficiency
- Low voltage operation + Multicore cluster & Extended RISC-V ISA

- 55 nm
- 40 nm
- 22 nm

Graph showing energy per operation vs. performance [MOPS] with different technologies (55 nm, 40 nm, 22 nm) highlighted.
**РИ5СУ: Xpulp ISA extension**

- Loop: Hw loop, LD/ST with post-increment
- Linear Algebra: single cycle MAC insns;
- DSP: 8-bit and 16-bit SIMD instructions
- Bit Manipulation: single cycle insert/extract
**ARM CMSIS-NN**

Open-Source Software library for QNN inference at the Edge

- **8-bit** and **16-bit** Fixed-point quantized weights and activations

- **Height-Width-Channel (HWC)** Data Layout

- **Matrix multiplication (GEMM)** based implementation of convolution and linear kernels

---

PULP-NN: Based on CMSIS-NN data layout

Target: **RV32IMCXpulp** Parallel Ultra-Low-Power Cluster

**8-bit** fixed-point network weights and pixels

- < 5% of accuracy loss w.r.t. 32-bit floating point (*)
- Supported by SIMD Xpulp instructions

PULP-NN: Xpulp ISA exploitation

8-bit Convolution

- RV32IMC
  - addi a0,a0,1
  - addi t1,t1,1
  - addi t3,t3,1
  - addi t4,t4,1
  - lbu a7,-1(a0)
  - lbu a6,-1(t4)
  - lbu a5,-1(t3)
  - lbu t5,-1(t1)
  - mul s1,a7,a6
  - mul a7,a7,a5
  - add s0,s0,s1
  - mul a6,a6,t5
  - add t0,t0,a7
  - mul a5,a5,t5
  - add t2,t2,a6
  - add t6,t6,a5
  - bne s5,a0,1c000bc

- RV32IMCXpulp
  - lp.setup
  - p.lw  w1, 4(a0!)
  - p.lw  w2, 4(a1!)
  - p.lw  x1, 4(a2!)
  - p.lw  x2, 4(a3!)
  - pv.sdotsp.b s1, w1, x1
  - pv.sdotsp.b s2, w1, x2
  - pv.sdotsp.b s3, w2, x1
  - pv.sdotsp.b s4, w2, x2
  - end

- 9x less instructions than RV32IMC

Pooling & ReLu
- HW loop
- LD/ST with post-increment
- 8-bit SIMD max, avg INSNS
PULP-NN: Exploring Data Reuse in the Register File

CMSIS-NN based Matrix Multiplication Layout: 2x2

\[ w1 \times x1 \times x2 = S1.\ S2 \]

PULP-NN Matrix Multiplication Layout: 4x2

\[ w1 \times x1 \times x2 = S1.\ S2 \]

RegisterFile of the RI5CY core: 32 general purpose registers

- 2x2: 43% utilization
- 4x2: 69% utilization

8-bit Convolution

More Data Reuse & Higher utilization of the RF

Peak Performance (8 cores)

- 2x2: 12.8 MAC/cyc
- 4x2: 15.5 MAC/cyc
Results: RV32IMCXpulp vs RV32IMC

8-bit Convolution Results

![Graph showing speedup comparison between RV32IMC and RV32IMCXpulp across different core counts.]

- **IDEAL SPEEDUP**: Overall speedup of 75x
- **PULP-NN** relies on **Xpulp**:
  - 8-bit SIMD ISA support
  - Zero-overhead Loop
  - LD/ST with post-increment
  - Parallelism
  - 32 32-bit registers in RF

Speedup w.r.t. RV32IMC (ISA does matter 😊)
Experimental Setup: 8-bit QNN trained on CIFAR-10

**CMSIS-NN**
- STM32L4 (90 nm)
- ARM Cortex-M4

**PULP-NN**
- STM32H7 (40 nm)
- ARM Cortex-M7

**PULP**
- GAP-8 (55 nm)
- PULP architecture

https://github.com/ARM-software/MLexamples/tree/master/cmsisnn-cifar10
Performance and energy efficiency on commercial MCUs

**LATENCY**

- GAP8: 30x
- STM32L4: 19.6x
- STM32H7: 19.6x

**PERFORMANCE**

- GAP8 170 Mhz: 36.8x
- GAP8 90 Mhz: 32.05x
- STM32L4 80 Mhz: 7.45x
- STM32H7 400 Mhz: 4.06x

**ENERGY EFFICIENCY**

- GAP8 170 Mhz (1.2 V): 14.1x
- GAP8 90 Mhz (1 V): 9.48x
- STM32L4 80 Mhz: 16.6x
- STM32H7 400 Mhz: 39.5x
Conclusion

- **PULP-NN**: Optimized library for QNN inference on PULP Clusters;
- By exploiting **Xpulp** we achieve a Speedup of **10x** (clock cycles) with respect to **RV32IMC** implementation;
- By exploiting fully the PULP cluster the Speedup increases up to **75x** (clock cycles) with respect to **RV32IMC**;
- Inferring a CIFAR-10 QNN on **GAP-8** running PULP-NN, we achieve **7.45x** higher performance and up to **39.5x** better energy efficiency with respect to a **high-end Cortex-M7** processor running CMSIS-NN;
- Also we achieve **14.1x** better energy-efficiency with respect to a **low-end Cortex-M4** processor.

**Resources:**

[https://github.com/pulp-platform/pulp-nn](https://github.com/pulp-platform/pulp-nn)

[https://github.com/pulp-platform/pulp](https://github.com/pulp-platform/pulp)