Swarnalata Panigrahy

+1 657 709 5570 swarnalp@uci.edu linkedin.com/in/swarai github.com/swarai-code

Education

M.S. in Electrical Engineering and Computer Science *Expected 2026*
University of California, Irvine Irvine, CA
Relevant Coursework: Computer Architecture, System-on-Chip Design, Languages and Compilers for Hardware Accelerators, Digital Signal Processing, Digital Image Processing, Random Processes and Optimizations

Technical Skills

Languages: C, C++, Python, MATLAB, SQL, Verilog, SystemVerilog
ML/AI Libraries: PyTorch, NumPy, PySR, ONNX, TorchScript, Scikit-learn, Hugging Face
Compiler/Build Tools: TVM, GCC/G++, CMake
Hardware Tools: Xilinx Vivado, Vitis HLS, PYNQ
Systems/Workflow: Linux, Git, Bash, Docker,Kubernetes.

Work Experience

Associate Engineer - Machine Learning (Recommendations and Insights) *June 2022 - May 2024*
Ingram Micro Mumbai, Maharashtra

- Built and deployed a production recommendation engine with 90% active catalog coverage across international e-commerce markets improving inference latency by 31% using batched prediction, result caching, and optimized output schemas.
- Validated model outputs, ranking quality, and data consistency across 28 countries through UAT and regression testing.
- Debugged production ML issues involving catalog features, localization mismatches, data inconsistencies, and incorrect recommendations.

Research Experience

Graduate Researcher, ML Systems and Edge AI Acceleration *September 2025 - Present*
CORSA LAB (Compiler Optimizations, Reconfigurable and Scalable Architectures) UC Irvine

- Working on NeuSym-HLS, a partial symbolic distillation and high-level synthesis flow for compressing and accelerating inference on edge hardware.
- Replacing selected layers of trained DNN models with compact analytic expressions generated through symbolic regression.
- Generating hardware accelerators for hybrid DNN-symbolic models using C/C++, Python, PyTorch, PySR,and HLS.
- Evaluating tradeoffs across inference accuracy, hardware resource usage, and latency for vision-model acceleration workloads.
- Studying hardware-aware compression beyond pruning and quantization to improve edge AI accelerator efficiency.

Projects

Mini Tensor Compiler for Neural Network Operators *Winter 2026*
Python, C++, NumPy, PyTorch

- Built a mini tensor compiler for neural network operators, lowering high-level tensor expressions into generated C++ loop kernels. Implemented layout-aware code generation for matmul, transpose, reductions, softmax, LayerNorm, ReLU, and INT8 quantize/dequantize operators.
- Added shape tracking and fusion support for add-ReLU and matmul-bias patterns to reduce temporary tensors and intermediate memory traffic.
- Validated generated kernels against NumPy and PyTorch references across 60+ tensor shapes using a maximum absolute FP32 error tolerance of 10^{-5}.

Tiled Matrix Multiplication Kernel Optimization *Fall 2025*
C/C++, Python, Vitis HLS

- Implemented naive, loop-reordered, and tiled matmul kernels, achieving 4.1× throughput improvement on 512×512 matrices.
- Swept 8×8 to 64×64 tile sizes to improve data reuse within a simulated 64 KB SRAM budget.
- Applied HLS loop pipelining and array partitioning, reducing estimated latency by 2.7× while validating outputs against NumPy.

DNN Acceleration on Kria KR260 FPGA *Fall 2025*
Vitis HLS, C/C++, Python, PYNQ, Xilinx Vivado

- Built a hardware/software DNN accelerator on Kria KR260, offloading convolution/fully connected layers to FPGA programmable logic with 3.2× speedup over ARM CPU baseline.
- Optimized target layers in Vitis HLS using loop pipelining and array partitioning while maintaining model accuracy within 1% of the software baseline.
- Developed PYNQ host code for buffer allocation, physical address passing, register setup, launch/polling, and debugging of layout, pointer, and timeout issues.