Project title:

From Neurocomputing Foundation to SNN Prototyping: An 8-Week Python-Based Project with an HDL Extension

Project Summary:

This project trains an undergraduate researcher to build an end-to-end spiking neural network (SNN) pipeline, progressing from neuromorphic fundamentals to hardware-verifiable implementation, aligned with NeuMat's materials, devices, and systems remit. Over 8 weeks, the student will implement and validate LIF/Exp-LIF neuron models in Python using structured tests and diagnostic voltage/spike/conductance plots. They will then build a compact SNN classifier with spike encoding and conductance-based synapses, add unsupervised STDP with winner-take-all competition, and quantify specialization, stability, runtime, and accuracy on a controlled task. The placement is supported by structured supervision, reproducible workflows, and a short fixed-point HDL capstone verified against Python.

Host institution/company and supervisors

Host: Queen Mary University of London (QMUL), School of Electronic Engineering and Computer Science
Supervisor: Dr Somayyeh Timarchi (Lecturer in Electronic Circuits and Systems)

Research quality

Edge AI for sensing and health applications require energy-efficient neuromorphic computation under tight latency, memory, and power constraints. Aligned with NeuMat's cross-stack focus on neuromorphic materials, devices, and systems for innovative AI hardware, this project develops a hardware-aware SNN prototyping and verification workflow that links algorithmic behaviour to implementable neuron/synapse building blocks. The novelty is twofold. (1) We will build a step-by-step, reproducible process to go from equations to a tested Python model, removing typical causes of wrong results (dt choice, reset/refractory rules, unit errors, and different synapse formulations). (2) We will make the work hardware-aware by adding practical constraints such as limited precision,

quantisation, and saturation, and testing how these change spiking and learning stability. The methodology is feasible for an 8-week UG studentship: (i) implement and validate LIF/Exp-LIF neurons in Python using a structured test suite and diagnostic plots; (ii) build a compact SNN classifier using spike encoding and conductance-based synapses; (iii) add unsupervised STDP with WTA competition and quantify specialization; and (iv) translate a core block to fixed-point VHDL/Verilog and verifying it. Expected outcomes include a curated benchmark/test set, publication-quality figures, and a reusable, validated codebase plus a verified HDL module to seed a PhD project.

Opportunity to learn across the stack

The project provides cross-stack training across materials, devices, and systems in neuromorphic computing.

Materials/device perspective: early sessions introduce how neuromorphic substrates motivate practical constraints, analogue conductance updates, variability, limited precision, and how these map to abstractions such as conductance-based synapses and bounded weights.

Systems perspective: using Python, the student will build an end-to-end SNN prototype: spike encoding (rate/temporal), conductance synapse dynamics (trace updates), neuron dynamics (LIF/Exp-LIF), a network update loop, and a readout for classification. They will implement standard test stimuli (step currents, pulse trains, repeated pattern inputs) and define readout rules (spike count / rate-based decision). A structured validation workflow (unit checks, dt sweeps, reset/refractory tests, and diagnostic voltage/spike/conductance plots) will verify correctness and isolate errors. The student will run controlled ablations (encoding choice, inhibition strength, learning on/off) and multi-seed repeats to quantify how modelling choices affect spike fidelity, learning stability, runtime, and accuracy, reporting results with reproducible logs and clear figures.

Cross-stack learning is delivered through structured collaboration with a SNN team via pair-programming, code reviews, and weekly team meetings where linking materials/device assumptions to system behaviour are discussed. A short capstone introduces hardware-aware thinking via fixed-point analysis and an HDL prototype verified against a Python model.

Research environment

The student will be embedded in an active SNN research group with structured supervision, inclusive day-to-day support, and clear milestones. The supervisor will oversee research direction, with mentors from the SNN team providing practical guidance. The student will have two scheduled meetings each week: (i) a technical progress session covering debugging, modelling choices, and experiment design; and (ii) a research-skills session on reading strategy, note-taking, writing, and presentation. Between meetings, rapid help is available via a team channel and ad-hoc check-ins. The training plan is a staged 8-week programme with weekly deliverables: validated neuron models, a working tiny SNN classifier, STDP learning results, and a fixed-point HDL capstone verified against a Python model. Onboarding in week 1 includes accounts, repository access, coding standards, experiment logging templates, and a safety/ethics briefing where relevant. Practical arrangements are lightweight (Python/Jupyter on standard PCs) with access to HDL tools for the final weeks. Travel/accommodation will be agreed early to ensure accessibility. Flexible/hybrid working and reasonable adjustments will be discussed and supported in week 1. The student will join group meetings and peer code reviews, and will finish with a short report and poster/talk. Weekly milestones are tracked in a Kanban board and checklist.