MASTER THESIS PROPOSAL

Informatics: Robotics and Intelligent Systems

Brage Wiseth, Studentnr. 651669	Author
Philipp Häfliger	Supervisor
Yngve Hafting	Co-Supervisor

2024-11-28

Overview

This project shall look into implementing an asynchronous spiking neural network on an FPGA.

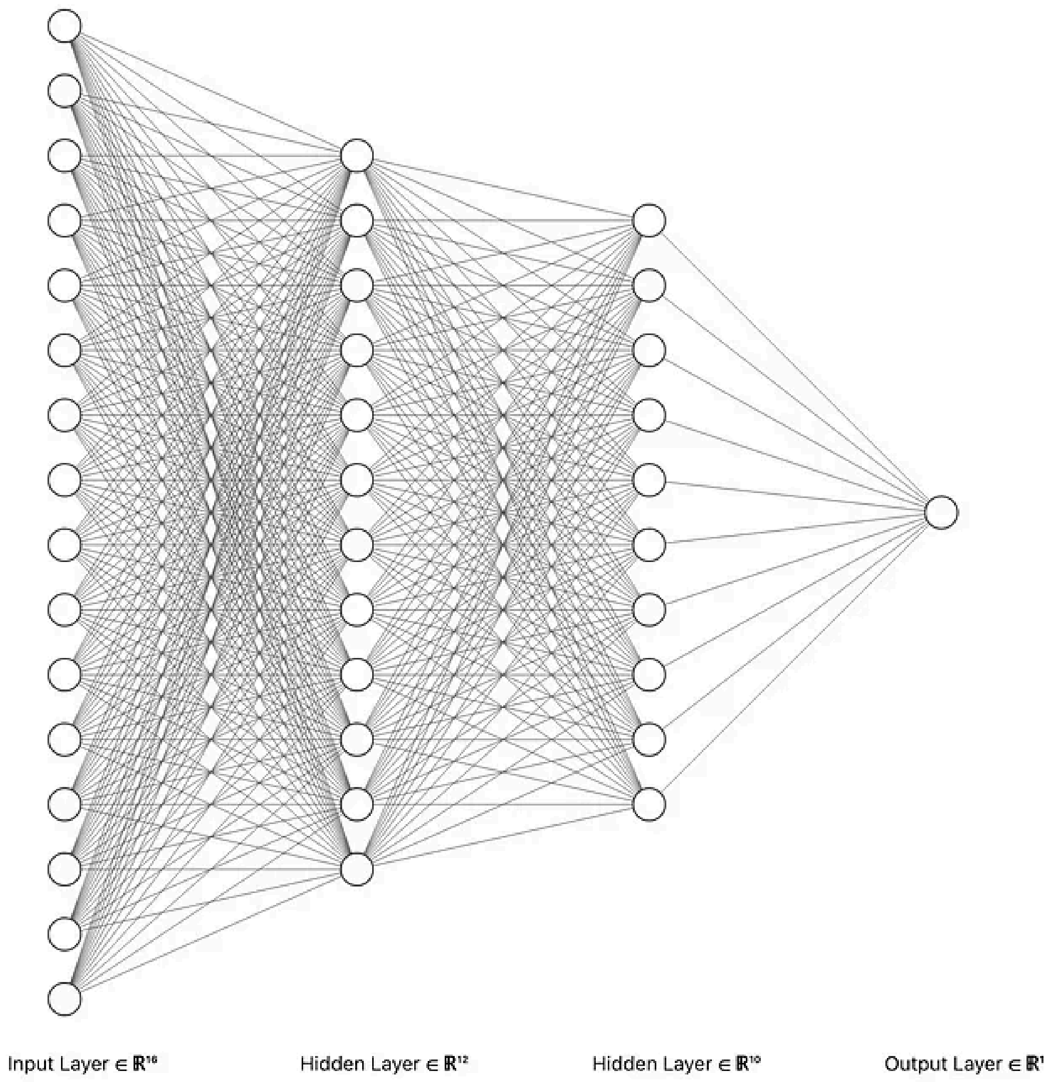
Background

Artificial Neural Networks (ANNs) have become a cornerstone of modern artificial intelligence, driving breakthroughs in diverse fields such as computer vision, natural language processing, and robotics. Despite their remarkable capabilities, conventional ANNs are computationally intensive, requiring significant energy and processing power. This limitation has motivated the exploration of more biologically inspired models, such as Spiking Neural Networks (SNNs). SNNs represent the third generation of neural network models, mimicking the behavior of biological neurons more closely than traditional ANNs. Unlike ANNs, which rely on continuous-valued activations and dense connectivity, SNNs process information through discrete spikes—short bursts of electrical activity. This temporal encoding enables event-driven computation, making SNNs inherently more efficient and well-suited for low-power applications. While software simulations of SNNs running on conventional hardware (e.g., CPUs and GPUs) have been extensively studied, the potential of custom hardware architectures remains underexplored. Such architectures can exploit the strengths of SNNs, such as sparse and asynchronous computation, to achieve real-time, power-efficient processing. Notable examples of neuromorphic hardware include IBM's TrueNorth, Intel's Loihi, and BrainChip's Akida, which demonstrate the feasibility of hardware-based SNN systems. We want to investigate FPGA implementations of asynchronous spiking neural networks (SNN). Specific for SNN is that the timing of the individual nerve pulses between neurons carries the information, rather than for instance the level or value of a signal. One specific encoding scheme is the 'time to first spike' encoding proposed by Simon Thorpe, which this project will employ at least as a starting point to emulate a more traditional artificial neural network (ANN) with a SNN. Here the delay of a pulse/spike after a reset of the spike emitting neuron indicates an analog value. A receiving neuron has to measure that delay to decode the value sent by the emitting neuron(s). In turn, the receiving neuron can do some computation with all these inputs, typically in an ANN a summation of all inputs and then a 'sigmoid' activation function on that sum to determine it's own next output, i.e. the delay for its own next spike. The network of interconnections bewteen neurons can be implemented using the so called address event representation (AER), i.e. an asynchronous communication protocol capable of sending events (i.e. the spikes) in real time in a network betweeen many emitters and receivers.

wants to be look into hardware that is even more specifically dedicated to efficient neural network computations.

Objectives

Using the above idea as a starting point the project will also investigate the most recent developments in SNNs and evaluate other approaches to implementing a classical ANN as an SNN.


Input Layer ∈ $\mathbb{R}^{16}$
Hidden Layer ∈ $\mathbb{R}^{12}$
Hidden Layer ∈ $\mathbb{R}^{10}$
Output Layer ∈ $\mathbb{R}^{1}$

There are two master projects associated with this activity that will require collaboration and coordination:

1. SNN digital electronics neuronal models
2. SNN digital electronics network implementation

THIS project will concern itself with the second research question. The goal is to find an appropriate spike event communication implementation for a network of spiking neuronal models. AER variants are one group of methods that are promising. An important aspect will be the degree of network flexibility that a chosen method permits. For example multi layer perceptrons with full connectivity (as depicted in Fig. 1) would demand relatively few resources for the network configuration, where spike events are simply conveyed to all neurons in a subsequent layer by a single inter-layer AER bus. Howevewr, MLPs do in turn demand a lot of resources for the implementation of synapses and synaptic weights, as each neuron has to decode every AE to find the associated weight. An alternative to fully connected MLPs can be convolutional neural networks (CNN) that are very widely used in AI for pattern recognition. In CNNs, weight patterns are identical for all neurons in a layer forming a 'convolution kernel'. This weight pattern is projected on an input field of neurons of the preceding layer and topologically shifted between neighbouring neurons. Thus each neuron responds to the same pattern of inputs, but at different locations on the input layer. Another intertwined question is whether the network actually uses rate coding or, for example, time to first spike encoding. This may or may not have an impact on the communication methodology used

Timeline

Period	Task	Delivarable
Spring 2025	Literature search, assessing SNN implementation variants, initial VHDL simulations	Evaluation report with selection of architectures
Fall 2025	VHDL code development for the most promising variants, HW testing	Test report
Spring 2026	Complete testing and simulation. Thesis writing, possibly paper	Thesis