Literature Review on Spiking Neural Networks (SNNs) and Hardware Implementations

1. Introduction

Spiking Neural Networks (SNNs) represent the third generation of neural networks, inspired by the functioning of biological neurons. Unlike traditional Artificial Neural Networks (ANNs) that rely on continuous activation values, SNNs communicate via discrete spikes, leveraging temporal dynamics to encode information. This biologically plausible mechanism enables SNNs to offer exceptional energy efficiency, making them ideal for edge computing and low-power applications.

Recent advances have focused on developing efficient training algorithms, such as surrogate gradient descent, and optimizing hardware implementations using FPGAs and custom neuromorphic chips. These innovations collectively aim to bridge the algorithm-hardware co-design gap and enable real-time SNN applications.

2. Core Components of SNNs

SNNs consist of spiking neuron models, synaptic connections, and temporal encoding mechanisms. The most widely used neuron model is the Leaky Integrate-and-Fire (LIF) model, which accumulates input current over time and emits a spike when a threshold is reached. Several works have proposed optimized LIF models for hardware, such as the asynchronous LIF neuron, which reduces clock dependency and supports scalable computation[1].

Encoding input data into spikes is a key challenge. Rate coding, temporal coding, and population coding are the primary schemes. For instance, the use of Poisson encoding has been demonstrated effectively in multiple FPGA designs for its simplicity and biological plausibility[1].

3. Training Techniques and Quantization

Training SNNs is difficult due to the non-differentiability of spike functions. Surrogate gradients provide an effective workaround, enabling backpropagation-compatible training. Post-training quantization has emerged as a major strategy for reducing memory and compute costs. IM-SNN, for example, demonstrates that ternary membrane potentials and quantized weights can drastically reduce resource usage with minimal accuracy loss, achieving up to 13× memory efficiency on datasets like CIFAR-10 and DVS-CIFAR10[2].

Similarly, SNNs targeting retinal prosthetics have leveraged PRANAS-generated spike data and 4-bit/8-bit quantized weights to meet tight power budgets while maintaining competitive accuracy (83.2% and 87.2%)[4].

4. Neuromorphic Hardware and Mapping Strategies

The successful deployment of SNNs on hardware platforms requires architectural innovations. Crossbar-based neuromorphic systems face routing congestion and synapse sharing issues. SpiNeMap addresses this by introducing clustering and placement algorithms that reduce energy consumption and latency by 45% and 21% respectively on the DYNAP-SE chip[3].

Meanwhile, NoC-based designs have also gained traction. One such system optimizes routers for multicast spike delivery using hybrid arbitration, reducing communication latency by 36.7% and improving throughput for large-scale SNNs[5].

5. Specialized Accelerators and Hybrid Architectures

Several works propose domain-specific accelerators. For DVS applications, an FPGA accelerator uses structured input sparsity and early-stopping to suppress redundant spikes, leading to reduced power and improved response time[8]. Another approach incorporates a CNN front-end with an SNN back-end in a hybrid SoC for image recognition, combining the feature extraction strength of CNNs with the temporal efficiency of SNNs[7].

A prediction-aided hardware architecture further explores spike suppression by anticipating inactive neuron states, reducing memory access energy by up to 42%[6].

6. On-Chip Learning and STDP Implementations

Unsupervised learning via Spike-Timing Dependent Plasticity (STDP) allows SNNs to adapt online. Two comparative studies explored hardware circuits for STDP: one showed that counter-based designs outperform shift registers in power and area efficiency[9], while another implemented an Address Event Representation (AER)-based STDP system suitable for large-scale integration, using pulse-width coding for efficient updates[10].

7. Conclusion and Future Directions

In summary, SNN research spans multiple levels of abstraction, from biological inspiration to hardware deployment. Trends such as low-precision representation,

hybrid learning architectures, efficient routing mechanisms, and asynchronous logic designs have propelled SNNs closer to practical edge deployment. Future work may explore dynamic reconfiguration, online learning integration, and neuromorphic multi-modal fusion systems.

References

[1] Event-driven Spiking Neural Networks using Asynchronous-Logic Network-on-Chip Routers in Field Programmable Gate Array (FPGA)

[2] IM-SNN: Memory-Efficient Spiking Neural Network with Low-Precision Membrane Potentials and Weights

[3] Mapping Spiking Neural Networks to Neuromorphic Hardware

[4] [RetinalSNN] Quantized Spiking Neural Networks on FPGA: An Application to Retinal Prosthetics

[5] A Novel NoC-based SNN Architecture with Optimized Routers

[6] A Prediction Scheme in Spiking Neural Network Hardware for Ultra-low Power Consumption

[7] An End-to-End SoC for Brain-Inspired CNN-SNN Hybrid Applications

[8] An FPGA-Based Event-Driven SNN Accelerator for DVS Applications With Structured Sparsity and Early-Stop

[9] Comparative Analysis of Digital STDP Learning Circuits Designed Using Counter and Shift Register

[10] Efficient Hardware Implementation of STDP for AER-Based Large-Scale SNN Neuromorphic System