SpiNNaker2: A Large-Scale Neuromorphic System for Event-Based and Asynchronous Machine Learning

Hector A. Gonzalez[1,2] **Jiaxin Huang**[2] **Florian Kelber**[1] **Khaleelulla Khan Nazeer**[1]
Tim Langer[1] **Chen Liu**[1] **Matthias Lohrmann**[1,2] **Amirhossein Rostami**[1] **Mark Schöne**[1]
Bernhard Vogginger[1] **Timo C. Wunderlich**[3] **Yexin Yan**[1] **Mahmoud Akl**[2] **Christian Mayr**[1]
[1]Technische Universität Dresden [2]SpiNNcloud Systems GmbH [3]Universitätsmedizin Berlin
{hector.gonzalez, jiaxin.huang, matthias.lohrmann, mahmoud.akl}@spinncloud.com,
{florian.kelber, khaleelulla.khan_nazeer, tim.langer, chen.liu,
amirhossein.rostami, mark.schoene, bernhard.vogginger, yexin.yan,
christian.mayr}@tu-dresden.de, timo.wunderlich@bih-charite.de

Abstract

The joint progress of artificial neural networks (ANNs) and domain specific hardware accelerators such as GPUs and TPUs took over many domains of machine learning research. This development is accompanied by a rapid growth of the required computational demands for larger models and more data. Concurrently, emerging properties of foundation models such as in-context learning drive new opportunities for machine learning applications. However, the computational cost of such applications is a limiting factor of the technology in data centers, and more importantly in mobile devices and edge systems. To mediate the energy footprint and non-trivial latency of contemporary systems, neuromorphic computing systems deeply integrate computational principles of neurobiological systems by leveraging low-power analog and digital technologies. SpiNNaker2 is a digital neuromorphic chip developed for scalable machine learning. The event-based and asynchronous design of SpiNNaker2 allows the composition of large-scale systems involving thousands of chips. This work features the operating principles of SpiNNaker2 systems, outlining the prototype of novel machine learning applications. These applications range from ANNs over bio-inspired spiking neural networks to generalized event-based neural networks. With the successful development and deployment of SpiNNaker2, we aim to facilitate the advancement of event-based and asynchronous algorithms for future generations of machine learning systems.

1 Introduction

Progress in machine learning and the availability of computational resources are tightly coupled. Especially, the breakthrough of deep learning can be attributed to the successful acceleration of deep neural network (DNN) training on large-scale data with graphics processing units (GPUs) [Krizhevsky et al., 2012]. Deep learning models continue to improve their task performance with the number of floating-point operations spent on training on a wide variety of tasks. Therefore, deep learning models have been scaled up at unprecedented speed up to the limit of compute availability [Sevilla et al., 2022, Thompson et al., 2022]. How far can we take the joint scaling of model parameters and hardware accelerators? Since the 90s, the peak hardware floating-point operations per second (FLOPS) grew by an average rate of 60000 per 20 years. However, the DRAM bandwidth and interconnect bandwidth only grew by an average rate of 100 and 30 per 20 years, respectively Gholami et al. [2021]. This development changes the requirements on algorithms and accelerators, with the focus shifting to a reduction of the overall communication in future machine learning systems.

Considering the highly scalable computational substrate of biological nervous systems, we observe that communication in the brain is a) locally dense but globally very sparse, b) temporally very sparse via

37th First Workshop on Machine Learning with New Compute Paradigms at NeurIPS 2023(MLNPCP 2023).

binary spike communication, and c) asynchronous. The field of neuromorphic computing develops algorithms and accelerators that leverage these principles to devise efficient and scalable systems. The core algorithmic concept to deliver these goals are spiking neural networks (SNNs). Although, many works promise high savings in energy consumption, achieving state-of-the-art performance on machine learning benchmarks proves to be difficult with SNNs [Tavanaei et al., 2019, Taherkhani et al., 2020, Yamazaki et al., 2022]. Recent works aim to bridge the performance gap between SNNs and DNNs. In particular, Woźniak et al. [2020] and Subramoney et al. [2023] combine deep neural network architectures with discontinuous step functions and state resets to avoid communication in their networks. We are in favor of broader research efforts on communication avoiding learning systems.

While accelerators propel deep learning research forward, this research in turn channels unprecedented investments into the development of accelerators tailored to the needs of the deep learning community. Such interplay between deep learning and accelerators creates a path dependency: The performance of dense DNNs drives investments in dense accelerators, which subsequently boosts the performance of dense DNNs, leading to further investment in these accelerators [Barham and Isard, 2019, Hooker, 2021]. This dynamic prompts the question of whether alternative paths exist. Are there fundamentally different combinations of algorithms and hardware that yield more efficient learning machines, assuming they benefit from the same technological advancements that dense DNNs and GPUs have experienced over the past decades?

To address the previous challenges, we present SpiNNaker2, a versatile accelerator for event-based and asynchronous machine learning (ML). SpiNNaker2 is a highly parallel system composed of asynchronously operating processing elements (PEs) interconnected by an efficient network on chip [Höppner et al., 2022]. Model designers working with SpiNNaker2 are neither limited by highly structured thread execution (compared to GPUs), constrained by procedural architectures lacking event-based awareness (e.g., pure DNN platforms [Prabhakar et al., 2022, Ignjatović et al., 2022]) nor do they have to restrict themselves to specific neuron implementations (compared to pure neuromorphic systems [Davies et al., 2018, Pehle et al., 2022]). More than 35k SpiNNaker2 chips were manufactured and are assembled into the world's largest brain-like supercomputer with about 5 million processing elements, which will be remotely available to interested researchers around the world. With this workshop contribution, we aim to accelerate the exploration of event-based and synchronous machine learning models as an alternative path to GPU-centric models.

2 The SpiNNaker2 System

SpiNNaker2 is a massively parallel compute system that can be scaled up from one standalone chip with 152 ARM Cortex M4F cores (e.g., complex edge systems as in Fig. 1) to millions of cores (e.g., supercomputer levels as in Fig. 1). Compared to conventional multiprocessing systems, there is no operating system that dynamically schedules compute tasks to cores with shared virtual memory. Instead, each core runs a small pre-compiled program on 128kB SRAM that executes simple tasks upon reception of *events*. Technically, the event-based processing is handled through ARM's interrupt controller that starts, pauses and resumes user functions depending on interrupts (IRQs), while utilizing a core sleep mode to save energy.

There are different means on how applications on different cores can communicate with each other. Within a SpiNNaker2 chip, the Network-on-Chip (NoC) offers high-speed point-to-point communication between cores and access to the off-chip DRAM. DMA (direct memory access) units in each core and the DRAM interfaces enable bulk transfer without stopping the cores. For scalable, system-wide communication, each chip has a dedicated SpiNNaker2 packet router, containing configurable routing tables, and 6 links to neighbour chips. On SpiNNaker2, different packet types with up to 128-bit payload allow the efficient communication between PEs, chips and boards. As an example, in SNN simulation multi-cast packets are used for the transmission of spikes where a 32-bit key represents the ID of the spiking neuron.

As the SpiNNaker2 system aims to speed up SNNs, DNNs or hybrid approaches, each core provides selected operation acceleration. These include exponential, logarithm, true and pseudo random number generation, as well as energy-efficient low-precision 8-bit/16-bit integer matrix multiplication and 2-dimensional convolution. The event-based nature of the system encourages autonomous and asynchronous execution between cores. We leverage this by dynamically adapting clock frequency and supply voltage of individual cores to further save energy. This power switching can be automated by being coupled to the application [Höppner et al., 2019, 2022, 2017, Yan, 2022]. Each chip has 2GB of DRAM to form a node and support memory-intensive DNN execution. PCBs with 48 of those nodes are used to build the 5-million core large-scale system. The chip connectivity in those boards rely on a hexagonal grid, assembling a torus-shaped network at a system level, which reduces the number of node hops compared to mesh interconnects.

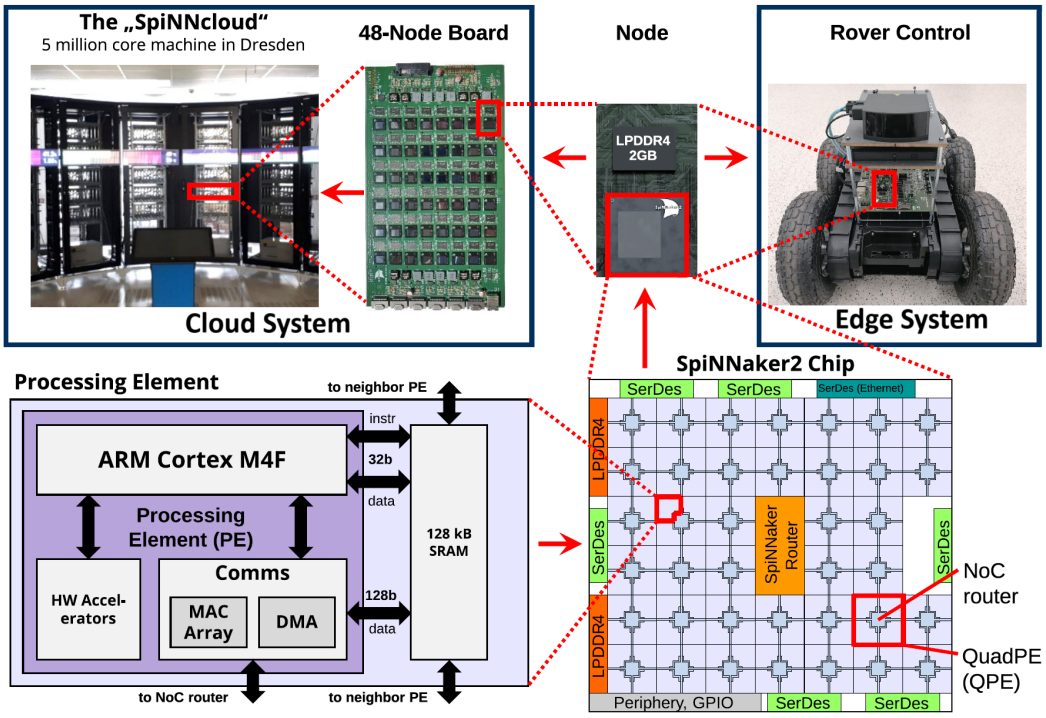

Figure 1: Overview of SpiNNaker2 architecture for the ”SpiNNcloud” cloud system and edge systems.

The main advantages of the SpiNNaker2-based systems then include its event-based nature, its native support for hybrid models beyond pure DNNs or SNNs (e.g., including symbolic models [Hammer, 2022]), and its high-speed infrastructure to scale toward arbitrary large systems without losing real-time capabilities. However, programming and operating those massively parallel systems is challenging. Compute problems such as SNNs, DNNs, graphical models, or sensor processing pipelines require small task partitioning, mapping, setting up event-based communication, data loading and result readings. In the following section we will show operating examples on single-chip SpiNNaker2 systems. Larger applications are planned via the operating principle from SpiNNaker1 ([Brown et al., 2014, Rowley et al., 2019]).

3 Applications

The first generation SpiNNaker system [Painkras et al., 2013, Furber et al., 2014] is used in 23 countries by more than 60 research groups ranging from neuromorphic computing to neuroscience [spi, 2023, Furber and Bogdan, 2020]. The SpiNNaker2 project expands the scope of algorithmic research to cover models from neuromorphic computing up to conventional machine learning models. In the following section, we present how ANNs and SNNs are mapped to SpiNNaker2, concluding with our initial work on event-based ANNs for both inference and training. Such hybrid approach takes the best of both worlds: The numerical simplicity and scalability of ANNs, combined with the ability of biology to reduce data flow and computation to the minimum necessary for a given task.

3.1 Artificial Neural Networks

A major use case of the SpiNNaker2 platform is the energy-efficient training and inference of ANNs, for which two approaches will be presented in sections 3.1.1 and 3.1.2.

3.1.1 Scheduling Operations for ANNs

Considering the large dimensions of current ANN architectures, the mapping of ANN layers onto PEs with limited storage and local accelerators becomes a challenge. Therefore, a scheduling approach for

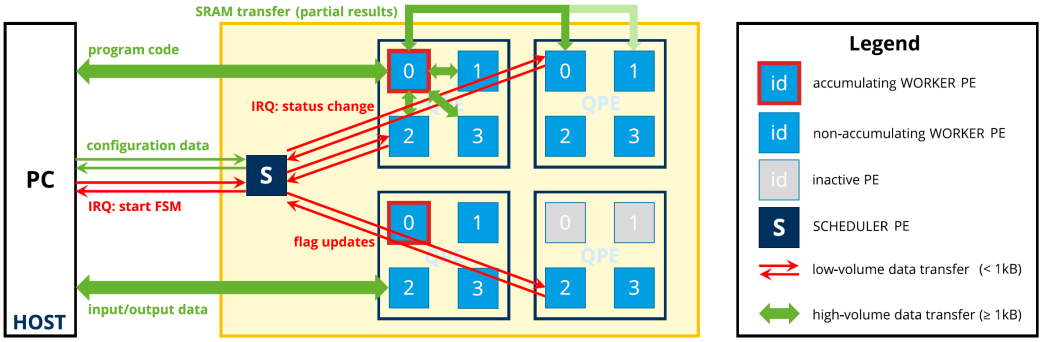

Figure 2: Control (red) and Data (green) flow of the graph execution with a host, *scheduler* and *worker* PEs.

SpiNNaker2 has been developed to distribute and efficiently compute layers across a multitude of PEs. The current scheduling approach (see Fig. 2) defines a single *scheduler* that coordinates the state transitions of *workers* using a finite state machine (FSM). After an initial interrupt request (IRQ) by the *host*, the asynchronous schedule execution is completely orchestrated on-chip by the *scheduler* PE to avoid slow chip-to-host communication via Ethernet. The schedule is executed statically (i.e. assignment of tasks to PEs and task execution sequence are pre-defined a-priori by *host*). The data exchange is organized decentrally and locally between PEs to avoid communication bottlenecks via the *scheduler*, while the control flow of status words is organized centralistically and hierarchically from the *scheduler* to the *workers* (see Fig. 2).

As an example, a distributed matrix-matrix multiplication requires the tiling of each input matrix into submatrices, the calculation of partial results by multiplication of the respective submatrices, the transfer of these partial results, and the accumulation of the corresponding partial results. The tiled input matrices will be transferred from the *host* to the *workers*. After an initial IRQ by the *scheduler*, the *workers* will start executing the multiplication of the submatrices. Having finished, each *worker* will write its updated flags to the *scheduler's* SRAM. In a loop, the *scheduler* is checking whether the state of any *worker* PE changed, and if positive, perform a state update. If a *worker* finished the matrix multiplication, it will either terminate, or fetch and locally accumulate the partial results of another *worker* within a pre-defined accumulation group. After the status update, the *scheduler* will send an IRQ towards the accumulating *worker* to initiate the fetching from another *worker's* SRAM. Once the updated operation flags are sent to the *scheduler*, the FSM update loop registers a flag change, updating the internal state of the specific *worker*, and triggering an IRQ combined with a flag update to the *worker*.

Future scheduling approaches will tailor full state-of-the-art ANNs such as large language models (LLMs) into the supercomputer fabric to reduce power consumption via the energy-proportional SpiNNaker2 features (e.g., event-based). Simulation frameworks such as Kelber et al. [2020] will be extended, including other *scheduler-worker* interactions (e.g., subgroup scheduling), as well as dynamic scheduling strategies.

3.1.2 Sparse-to-Sparse Training: Deep Rewiring

The sampling behavior observed in biology Kappel et al. [2018], Yan et al. [2019] demonstrates memory-efficient learning. Analogously, a learning algorithm known as deep rewiring maintains a consistent level of sparsity by dynamically disconnecting insignificant synapses and randomly establishing connections elsewhere throughout the entire training process. This approach facilitates learning in memory-constrained environments, particularly on edge devices.

The application of deep rewiring to SpiNNaker2 prototype chip Liu et al. [2018] showcases a highly sparse stochastic gradient descent (SGD) training from scratch. Remarkably, it achieves a 96.6% accuracy on the MNIST dataset for handwritten digits with 3 dense layers involving 410 neurons, while operating within a tight memory constraint of 64 kB and maintaining a connectivity of 1.3%. Time profiling reveals that the rewiring step dominates the computation time. This impact is mitigated by adjusting the rewiring frequency across iterations and leveraging the exponential accelerator and random number generator integrated in SpiNNaker2. The training time of the 4-core deep rewiring on SpiNNaker2 is on par with that of a standard X86 CPU (Intel i5-6500), while the energy consumption is substantially reduced by two orders of magnitude. These findings underscore the potential of incorporating bio-inspired learning algorithms, such as deep rewiring, into SpiNNaker2, signaling a paradigm shift towards a more energy-efficient computation.

3.2 Spiking Neural Networks

Despite SpiNNaker2 innovating with event-based ANNs, its performance in bio-inspired SNNs remains being unique as it operates at scales that are not reachable by other neuromorphic systems. SNNs have the potential to be more efficient than conventional ANNs because they mimic the brain mechanism of sparse communication between neurons and only transmit spikes when necessary, which greatly reduces the energy footprint. In general, a series of transformation steps, such as parsing and preprocessing SNN model information, are required to build the bridge between the trained SNN model and inference execution on SpiNNaker2. The transformation steps start from the specifically trained SNN model (see section 3.3) or from an ANN-converted SNN model (e.g. using SNN toolbox Rueckauer et al. [2017]). Then the SNN model is interpreted into an application graph, presenting the spike flow path among neuron populations. Each population is further split into one or several sub-populations to fit the SRAM resource of each PE. All the sub-populations and the corresponding projections form a machine graph. The connection relations of these sub-populations contribute to the generation of a routing table. Finally, these transformed results are presented with c files to be loaded on SpiNNaker2 along with an input spike train before execution. This mapping framework enables large-scale SNN simulation on multiple PEs of SpiNNaker2.

During runtime, the event-based synapse processing, time-triggered neuron state update and spike-based communication are applied to SNN execution. An event is a spike from a pre-synaptic neuron of another PE, and this spike triggers the synapses in the current PE to start processing it. The neuron states of the current PE are updated by programmable neuron models at a predefined regular time interval. Then the generated spikes are sent to the PEs with the post-synaptic neurons. This process has been showcased with synfire chain model, bursting network, an asynchronous irregular firing network from Höppner et al. [2019] and radar gesture recognition demonstration from Huang et al. [2022b] Huang et al. [2022a] as examples. Besides, such process can be accelerated by exploiting the MAC array on SpiNNaker2, with the spatial-temporal performance improved to some extent Huang et al. [2023b] Huang et al. [2023a]. Very recently, the neuromorphic intermediate representation (NIR, Pedersen et al. [2023]) was introduced offering an exchange format for SNN. Currently NIR is supported by 7 neuromorphic simulators and 4 hardware platforms. It allows to train deep SNN in frameworks like snnTorch Eshraghian et al. [2023] or Norse [Pehle and Pedersen, 2021] and deploy them on SpiNNaker2 using py-spinnaker2 Vogginger et al. [2023].

3.3 Event-Based Artificial Neural Networks

3.3.1 Event-Based Gated Recurrent Unit

Subramoney et al. [2023] propose the Event-based Gated Recurrent Unit (EGRU) to combine the energy efficiency of SNNs and the performance of ANNs. Therefore, EGRU employs a bio-inspired activity sparsity mechanism that allows the units to emit discrete and sparse-in-time events for communication. Since events are sent sparingly, this leads to substantial computational savings during training and inference. To validate these claims, a 2 layer EGRU model is implemented and distributed it over 128 PEs on a single SpiNNaker2 chip. The model is trained on a GPU similar to Subramoney et al. [2023] on the DVS gesture recognition task [Amir et al.] and the weights are transferred to the SpiNNaker2 PE memory. A CNN head is used to preprocess the dataset that is stored on the DRAM to be loaded onto local memory for every sample. Fig. 3 shows the energy-per-timestep measurements, normalized over 18 samples of DVS gestures. The comparison is made with inference on 2 GPUs (Nvidia A100 & GTX1070Ti), noticing a lower energy consumption on SpiNNaker2 compared to GPU inference. SpiNNaker2 displays a remarkable performance at batch-one or real-time conditions, but its energy remains constant for larger batches. In terms of EGRU training, Subramoney et al. [2023] also shows an event-based learning rule similar to EventProp in the limit of continuous time. The cell state equation of the GRU can be viewed as the Euler discretization of a continuous time dynamical system. Based on the theory of adjoint sensitivity analysis in EventProp (see section 3.3.2), Subramoney et al. [2023] derives the adjoint equations for a GRU system under discrete state transitions triggered by input events. Such a system is similarly suited for a SpiNNaker2 implementation.

3.3.2 EventProp: Event-Based Backpropagation

EventProp Wunderlich and Pehle [2021] is a learning algorithm for event-based backpropagation that computes exact gradients in SNNs while retaining the temporal sparsity of spike-based communication during the backward pass. In a *proof-of-concept* demonstration, we show that SpiNNaker2 can implement EventProp for multi-layer feed-forward SNNs. Every SpiNNaker2 core implements a clock-driven simulation of a layer of leaky integrate-and-fire neurons. During the forward pass, spike events are

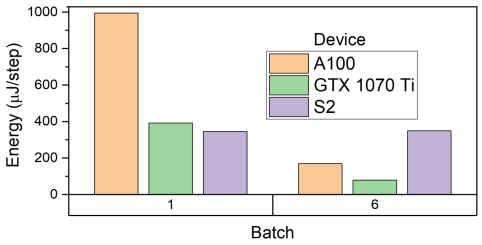

Figure 3: Energy per timestep for EGRU.

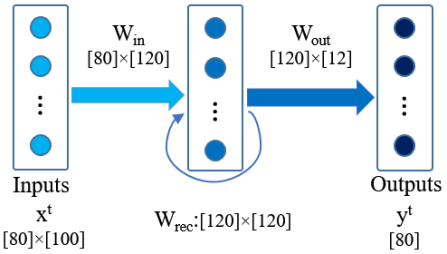

Figure 4: SRNN used in the e-prop algorithm.

distributed across cores using multi-cast packets routed by the network-on-chip. During the backward pass, spike events are distributed in reverse and carry error signals contained in the 128-bit payload of the multi-cast packets. After the backward pass, a designated control core collects gradients from cores and computes weight updates using the Adam optimizer Kingma and Ba [2015]. This allows for batch-parallel processing by accumulating gradients from different copies of the same layer, computed on different cores. The viability of this approach is demonstrated with a time-to-first-spike loss function, a single hidden layer network, and by achieving latency-encoded parallel training. The results show that SpiNNaker2 supports event-based backpropagation through multi-cast event routing, along with batch execution.

3.3.3 E-prop: Biologically-Inspired Learning

Eligibility propagation (e-prop), a biologically-inspired online learning rule for Spiking Recurrent Neural Networks (SRNNs), serves as an alternative to Back Propagation Through Time (BPTT) [Bellec et al., 2020]. The e-prop gradient at any given time step is independent of subsequent step information, enhancing memory efficiency and suitability for online learning. As validation, a 3-layer neural network (Fig. 4) classifying the 12-category Google Speech Command dataset was implemented in Rostami et al. [2022]. While BPTT required 859KB of memory, e-prop used only 682KB (20% less). For parallelization, synapses were evenly distributed across the PEs, allowing for local gradient computation at each time step and requiring only the spike transmission. Nevertheless, in the last time step, the output error is broadcast to other PEs for weight updating. The results show that SpiNNaker2 enables efficient spike transmission in algorithms such as e-prop, along with high test accuracies (i.e., 91.12%) under real-time (batch-one) conditions.

4 Discussion and Outlook

The proposed SpiNNaker2 system enables large-scale simulation of event-based and asynchronous machine learning systems, two essential properties of scalable computational systems. The system deviates from the major direction focused by the deep learning community with the perspective to allow gains in energy efficiency and latency at scale. Turning from mostly dense and synchronous processing to event-based and asynchronous processing sets significantly different requirements for the development of learning algorithms. Despite the programming challenges, SpiNNaker2 paves the path for future fully event-based supercomputers with a neglectable Amdahl limit and an energy-proportional operation beyond DNNs.

Among the projects described in this paper, EventProp and the event-based learning rule for EGRU demonstrate error propagation in event-based and potentially asynchronous neural networks. E-prop shows the utilization of locally restricted error signals to save communication and hence energy during training. In addition, sparse-to-sparse training in the Deep Rewiring approach shows how to improve further the efficiency.

With the aim of driving research in this algorithmic front, the 5-million core system in Dresden, Germany, will grant access to researchers keen on exploring these challenges with us. Such supercomputer finds applications in energy-efficient LLMs by leveraging for example event-based recurrent structures in the recent resurgence of RNNs against transformers at language modeling tasks [Dao et al., 2022, Peng et al., 2023, Sun et al., 2023]. Furthermore, the applications also include but are not limited to the new development of event-based ML, the large-scale deployment of hybrid models such as NARS [Hammer, 2022], complex brain simulation, the utilization of the massive parallelism in probabilistic computing and distributed drug discovery. Systems with arbitrary sizes are also commercially available in [Spi, 2023].

Acknowledgements

MS is fully funded by the Bosch Research Foundation. The authors gratefully acknowledge the GWK support for funding this project by providing computing time through the Center for Information Services and HPC (ZIH) at TU Dresden. The authors also acknowledge the EIC Transition under the "SpiNNode" project (grant number 101112987), and the support by the German BMBF (Joint project 6G-life, ID: 16KISK001K and DAAD project SECAI, ID: 57616814). This work was partially funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) under the contract 01MN23004F.

References

SpiNNcloud Systems GmbH, 2023. URL https://spinncloud.com/. Accessed on October 3, 2023.

SpiNNworld: SpiNNaker presence worldwide, 2023. URL https://www.google.com/maps/d/u/0/edit?mid=1jrbV2OVaBFqGlVYMxerSh0Pcexd_wznQ&ll=1.5170674512027844%2C0&z=2. Accessed on October 3, 2023.

Arnon Amir, Brian Taba, David Berg, Timothy Melano, Jeffrey McKinstry, Carmelo Di Nolfo, Tapan Nayak, Alexander Andreopoulos, Guillaume Garreau, Marcela Mendoza, Jeff Kusnitz, Michael Debole, Steve Esser, Tobi Delbruck, Myron Flickner, and Dharmendra Modha. A Low Power, Fully Event-Based Gesture Recognition System. page 10.

Paul Barham and Michael Isard. Machine learning systems are stuck in a rut. In *Proceedings of the Workshop on Hot Topics in Operating Systems*, HotOS '19, page 177–183, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367271. doi: 10.1145/3317550.3321441. URL https://doi.org/10.1145/3317550.3321441.

Guillaume Bellec, Franz Scherr, Anand Subramoney, Elias Hajek, Darjan Salaj, Robert Legenstein, and Wolfgang Maass. A solution to the learning dilemma for recurrent networks of spiking neurons. *Nature Communications*, 11(1):3625, July 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-17236-y.

Andrew D Brown, Steve B Furber, Jeffrey S Reeve, Jim D Garside, Kier J Dugan, Luis A Plana, and Steve Temple. Spinnaker—programming model. *IEEE Transactions on Computers*, 64(6):1769–1782, 2014.

Tri Dao, Daniel Y Fu, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *arXiv preprint arXiv:2212.14052*, 2022.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1):82–99, 2018.

Jason K Eshraghian, Max Ward, Emre Neftci, Xinxin Wang, Gregor Lenz, Girish Dwivedi, Mohammed Bennamoun, Doo Seok Jeong, and Wei D Lu. Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9):1016–1054, 2023.

Steve Furber and Petrut Bogdan, editors. *SpiNNaker: A Spiking Neural Network Architecture*. now publishers, Boston-Delft, 2020. doi: 10.1561/9781680836523. URL http://dx.doi.org/10.1561/9781680836523.

Steve B. Furber, Francesco Galluppi, Steve Temple, and Luis A. Plana. The spinnaker project. *Proceedings of the IEEE*, 102(5):652–665, 2014. doi: 10.1109/JPROC.2014.2304638.

Amir Gholami, Zhewei Yao, Sehoon Kim, Michael W Mahoney, and Kurt Keutzer. Ai and memory wall. *RiseLab Medium Post*, 2021. URL https://github.com/amirgholami/ai_and_memory_wall.

P Hammer. Reasoning-learning systems based on non-axiomatic reasoning system theory, vol. 192. 2022.

Sara Hooker. The hardware lottery. *Commun. ACM*, 64(12):58–65, nov 2021. ISSN 0001-0782. doi: 10.1145/3467017. URL https://doi.org/10.1145/3467017.

Jiaxin Huang, Pascal Gerhards, Felix Kreutz, Bernhard Vogginger, Florian Kelber, Daniel Scholz, Klaus Knobloch, and Christian Georg Mayr. Spiking neural network based real-time radar gesture recognition live demonstration. In *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 500–500, 2022a. doi: 10.1109/AICAS54282.2022.9869943.

Jiaxin Huang, Bernhard Vogginger, Pascal Gerhards, Felix Kreutz, Florian Kelber, Daniel Scholz, Klaus Knobloch, and Christian Georg Mayr. Real-time radar gesture classification with spiking neural network on spinnaker 2 prototype. In *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 362–365, 2022b. doi: 10.1109/AICAS54282.2022.9869987.

Jiaxin Huang, Florian Kelber, Bernhard Vogginger, Chen Liu, Felix Kreutz, Pascal Gerhards, Daniel Scholz, Klaus Knobloch, and Christian G. Mayr. Efficient snn multi-cores mac array acceleration on spinnaker 2. *Frontiers in Neuroscience*, 17, 2023a. ISSN 1662-453X. doi: 10.3389/fnins.2023.1223262. URL https://www.frontiersin.org/articles/10.3389/fnins.2023.1223262.

Jiaxin Huang, Florian Kelber, Bernhard Vogginger, Binyi Wu, Felix Kreutz, Pascal Gerhards, Daniel Scholz, Klaus Knobloch, and Christian Georg Mayr. Efficient algorithms for accelerating spiking neural networks on mac array of spinnaker 2. In *2023 IEEE 5th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, pages 1–5, 2023b. doi: 10.1109/AICAS57966.2023.10168559.

Sebastian Höppner, Yexin Yan, Bernhard Vogginger, Andreas Dixius, Johannes Partzsch, Prateek Joshi, Felix Neumärker, Stephan Hartmann, Stefan Schiefer, Stefan Scholze, Georg Ellguth, Love Cederstroem, Matthias Eberlein, Christian Mayr, Steve Temple, Luis Plana, Jim Garside, Simon Davison, David R. Lester, and Steve Furber. Live demonstration: Dynamic voltage and frequency scaling for neuromorphic many-core systems. In *2017 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–1, 2017. doi: 10.1109/ISCAS.2017.8050396.

Sebastian Höppner, Bernhard Vogginger, Yexin Yan, Andreas Dixius, Stefan Scholze, Johannes Partzsch, Felix Neumärker, Stephan Hartmann, Stefan Schiefer, Georg Ellguth, Love Cederstroem, Luis A. Plana, Jim Garside, Steve Furber, and Christian Mayr. Dynamic power management for neuromorphic many-core systems. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 66(8):2973–2986, 2019. doi: 10.1109/TCSI.2019.2911898.

Sebastian Höppner, Yexin Yan, Andreas Dixius, Stefan Scholze, Johannes Partzsch, Marco Stolba, Florian Kelber, Bernhard Vogginger, Felix Neumärker, Georg Ellguth, Stephan Hartmann, Stefan Schiefer, Thomas Hocker, Dennis Walter, Genting Liu, Jim Garside, Steve Furber, and Christian Mayr. The spinnaker 2 processing element architecture for hybrid digital neuromorphic computing, 2022.

Drago Ignjatović, Daniel W. Bailey, and Ljubisa Bajić. The wormhole ai training processor. In *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, volume 65, pages 356–358, 2022. doi: 10.1109/ISSCC42614.2022.9731633.

David Kappel, Robert Legenstein, Stefan Habenschuss, Michael Hsieh, and Wolfgang Maass. A dynamic connectome supports the emergence of stable computational function of neural circuits through reward-based learning. *eNeuro*, 5(2), 2018. doi: 10.1523/ENEURO.0301-17.2018. URL https://www.eneuro.org/content/5/2/ENEURO.0301-17.2018.

Florian Kelber, Binyi Wu, Bernhard Vogginger, Johannes Partzsch, Chen Liu, Marco Stolba, and Christian Mayr. Mapping deep neural networks on spinnaker2. In *Proceedings of the 2020 Annual Neuro-Inspired Computational Elements Workshop*, NICE '20, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450377188. doi: 10.1145/3381755.3381778. URL https://doi.org/10.1145/3381755.3381778.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Chen Liu, Guillaume Bellec, Bernhard Vogginger, David Kappel, Johannes Partzsch, Felix Neumärker, Sebastian Höppner, Wolfgang Maass, Steve B. Furber, Robert Legenstein, and Christian G. Mayr. Memory-efficient deep learning on a spinnaker 2 prototype. *Frontiers in Neuroscience*, 12, 2018. ISSN 1662-453X. doi: 10.3389/fnins.2018.00840. URL https://www.frontiersin.org/articles/10.3389/fnins.2018.00840.

Eustace Painkras, Luis A. Plana, Jim Garside, Steve Temple, Francesco Galluppi, Cameron Patterson, David R. Lester, Andrew D. Brown, and Steve B. Furber. Spinnaker: A 1-w 18-core system-on-chip for massively-parallel neural network simulation. *IEEE Journal of Solid-State Circuits*, 48(8):1943–1953, 2013. doi: 10.1109/JSSC.2013.2259038.

Jens E. Pedersen, Steven Abreu, Matthias Jobst, Gregor Lenz, Vittorio Fra, Felix C. Bauer, Dylan R. Muir, Peng Zhou, Bernhard Vogginger, Kade Heckel, Gianvito Urgese, Sadasivan Shankar, Terrence C. Stewart, Jason K. Eshraghian, and Sadique Sheik. Neuromorphic intermediate representation: A unified instruction set for interoperable brain-inspired computing, 2023.

Christian Pehle and Jens Egholm Pedersen. Norse - A deep learning library for spiking neural networks, January 2021. URL `https://doi.org/10.5281/zenodo.4422025`. Documentation: https://norse.ai/docs/.

Christian Pehle, Sebastian Billaudelle, Benjamin Cramer, Jakob Kaiser, Korbinian Schreiber, Yannik Stradmann, Johannes Weis, Aron Leibfried, Eric Müller, and Johannes Schemmel. The brainscales-2 accelerated neuromorphic system with hybrid plasticity. *Frontiers in Neuroscience*, 16:795876, 2022.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, Xuzheng He, Haowen Hou, Przemyslaw Kazienko, Jan Kocon, Jiaming Kong, Bartlomiej Koptyra, Hayden Lau, Krishna Sri Ipsit Mantri, Ferdinand Mom, Atsushi Saito, Xiangru Tang, Bolun Wang, Johan S. Wind, Stansilaw Wozniak, Ruichong Zhang, Zhenyuan Zhang, Qihang Zhao, Peng Zhou, Jian Zhu, and Rui-Jie Zhu. Rwkv: Reinventing rnns for the transformer era, 2023.

Raghu Prabhakar, Sumti Jairath, and Jinuk Luke Shin. Sambanova sn10 rdu: A 7nm dataflow architecture to accelerate software 2.0. In *2022 IEEE International Solid- State Circuits Conference (ISSCC)*, volume 65, pages 350–352, 2022. doi: 10.1109/ISSCC42614.2022.9731612.

Amirhossein Rostami, Bernhard Vogginger, Yexin Yan, and Christian G. Mayr. E-prop on spinnaker 2: Exploring online learning in spiking rnns on neuromorphic hardware. *Frontiers in Neuroscience*, 16, 2022. ISSN 1662-453X. doi: 10.3389/fnins.2022.1018006. URL `https://www.frontiersin.org/articles/10.3389/fnins.2022.1018006`.

Andrew GD Rowley, Christian Brenninkmeijer, Simon Davidson, Donal Fellows, Andrew Gait, David R Lester, Luis A Plana, Oliver Rhodes, Alan B Stokes, and Steve B Furber. Spinntools: the execution engine for the spinnaker platform. *Frontiers in neuroscience*, 13:231, 2019.

Bodo Rueckauer, Iulia-Alexandra Lungu, Yuhuang Hu, Michael Pfeiffer, and Shih-Chii Liu. Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in Neuroscience*, 11, 2017. ISSN 1662-453X. doi: 10.3389/fnins.2017.00682. URL `https://www.frontiersin.org/articles/10.3389/fnins.2017.00682`.

Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. Compute trends across three eras of machine learning. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2022. doi: 10.1109/IJCNN55064.2022.9891914.

Anand Subramoney, Khaleelulla Khan Nazeer, Mark Schöne, Christian Mayr, and David Kappel. Efficient recurrent architectures through activity sparsity and sparse back-propagation through time. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=lJdOlWg8td`.

Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models, 2023.

Aboozar Taherkhani, Ammar Belatreche, Yuhua Li, Georgina Cosma, Liam P. Maguire, and T.M. McGinnity. A review of learning in biologically plausible spiking neural networks. *Neural Networks*, 122:253–272, 2020. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2019.09.036. URL `https://www.sciencedirect.com/science/article/pii/S0893608019303181`.

Amirhossein Tavanaei, Masoud Ghodrati, Saeed Reza Kheradpisheh, Timothée Masquelier, and Anthony Maida. Deep learning in spiking neural networks. *Neural Networks*, 111: 47–63, 2019. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2018.12.002. URL `https://www.sciencedirect.com/science/article/pii/S0893608018303332`.

Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. The computational limits of deep learning, 2022.

Bernhard Vogginger, Florian Kelber, Matthias Jobst, Yexin Yan, Pascal Gerhards, Martin Weih, and Mahmoud Akl. py-spinnaker2, November 2023. URL https://doi.org/10.5281/zenodo.10202110.

Stanisław Woźniak, Angeliki Pantazi, Thomas Bohnstingl, and Evangelos Eleftheriou. Deep learning incorporating biologically inspired neural dynamics and in-memory computing. *Nature Machine Intelligence*, 2(6):325–336, Jun 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-0187-0. URL https://doi.org/10.1038/s42256-020-0187-0.

Timo C. Wunderlich and Christian Pehle. Event-based backpropagation can compute exact gradients for spiking neural networks. *Scientific Reports*, 11(1):12829, Jun 2021. ISSN 2045-2322. doi: 10.1038/s41598-021-91786-z. URL https://doi.org/10.1038/s41598-021-91786-z.

Kashu Yamazaki, Viet-Khoa Vo-Ho, Darshan Bulsara, and Ngan Le. Spiking neural networks and their applications: A review. *Brain Sciences*, 12(7), 2022. ISSN 2076-3425. doi: 10.3390/brainsci12070863. URL https://www.mdpi.com/2076-3425/12/7/863.

Yexin Yan. *Implementation of bioinspired algorithms on the neuromorphic VLSI system SpiNNaker 2*. PhD thesis, TU Dresden, 2022.

Yexin Yan, David Kappel, Felix Neumärker, Johannes Partzsch, Bernhard Vogginger, Sebastian Höppner, Steve Furber, Wolfgang Maass, Robert Legenstein, and Christian Mayr. Efficient reward-based structural plasticity on a spinnaker 2 prototype. *IEEE Transactions on Biomedical Circuits and Systems*, 13(3): 579–591, 2019. doi: 10.1109/TBCAS.2019.2906401.