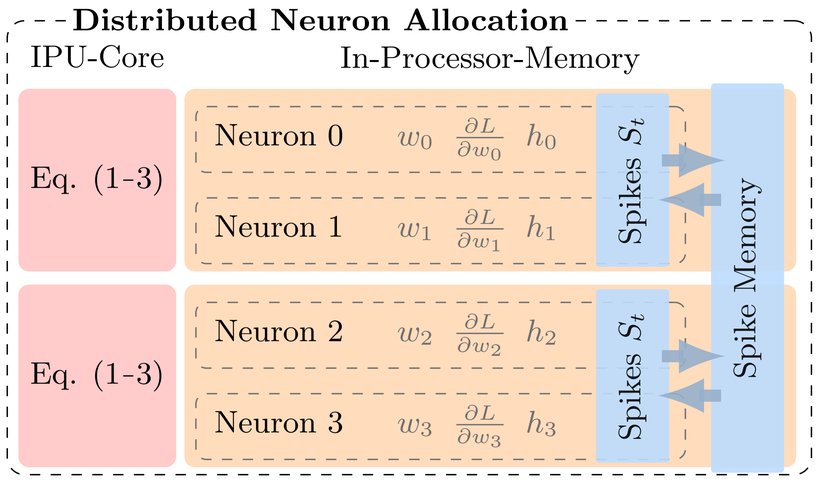
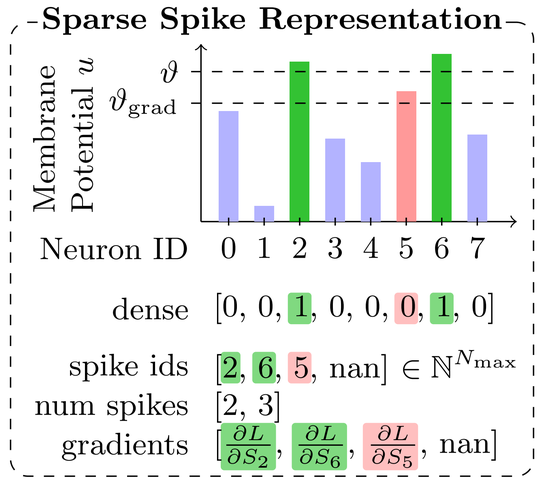
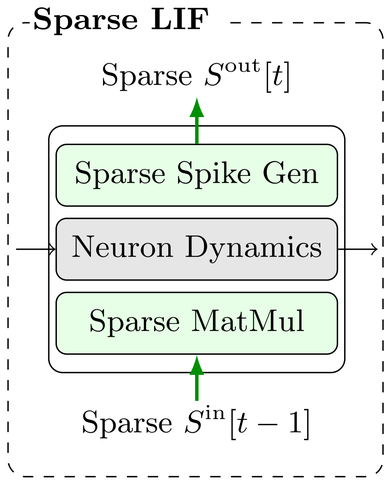
Exploiting Sparsity for Accelerated SNN Training on Graphcore IPUs

Jan Finkbeiner, Emre Neftci
jan.finkbeiner@fz-juelich.de, e.neftci@fz-juelich.de







- Time discretized dynamics, of Leaky Integrate and Fire (LIF) neurons for the membrane potential u:

$$u_i[t+1] = \alpha\, u_i[t](1 - S_i^{\text{out}}[t]) + \frac{1}{C}\, I_i[t] \quad (1)$$

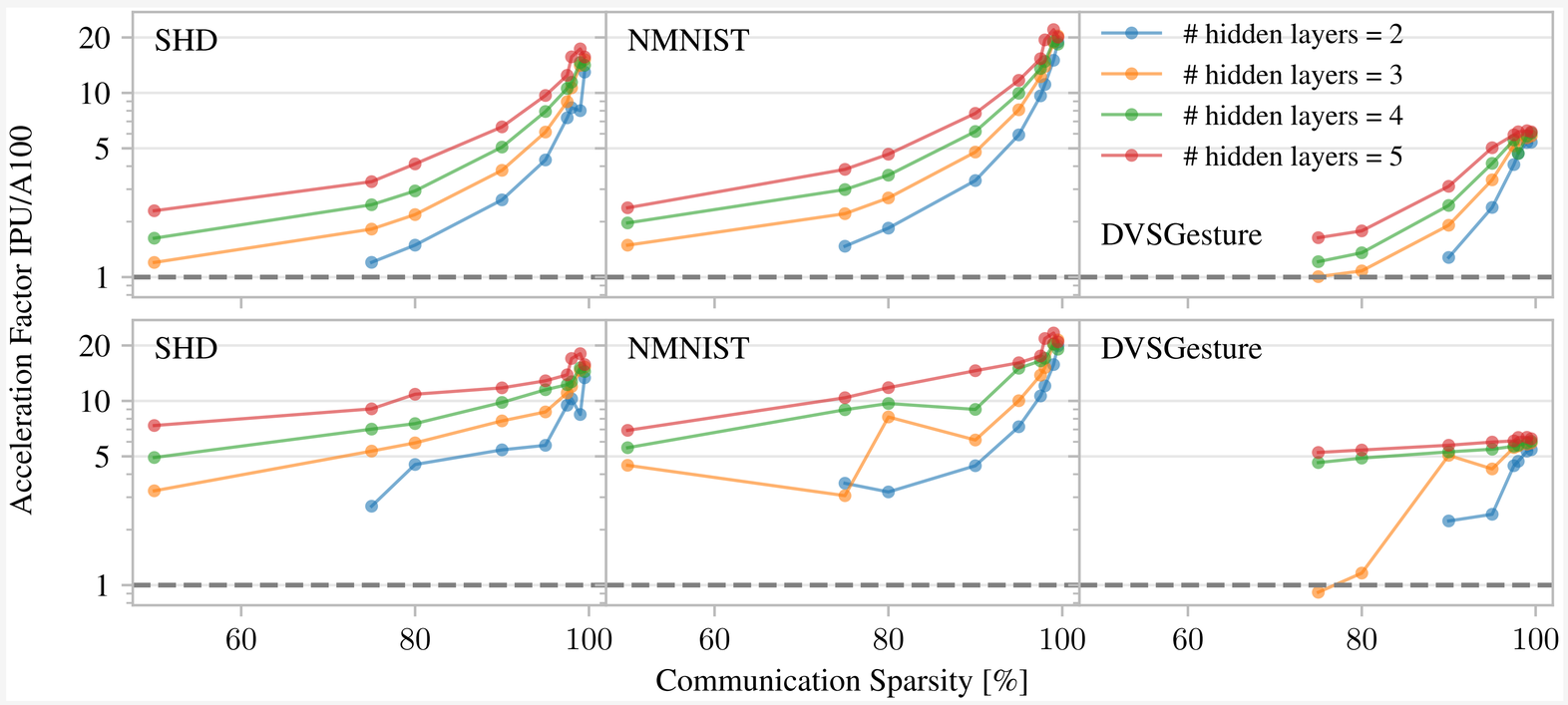
$$S_i^{\text{out}}[t+1] = \Theta(u_i[t+1] - \vartheta_i) \quad (2)$$

$$I_i[t+1] = \sum w_{ij}\, S_j^{\text{in}}[t+1] \quad (3)$$

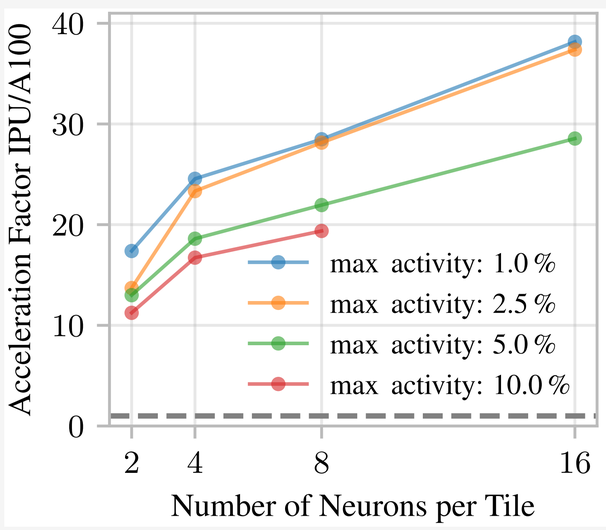
- Compressed sparse spike representation for both efficient communication and computation (Eq. 2-3)
- Distributed neuron allocation on the Graphcore IPU exploiting its distributed memory architecture composed of 1472 IPU-Cores with dedicated local In-Processor-Memory (628kB of SRAM per IPU-Core)

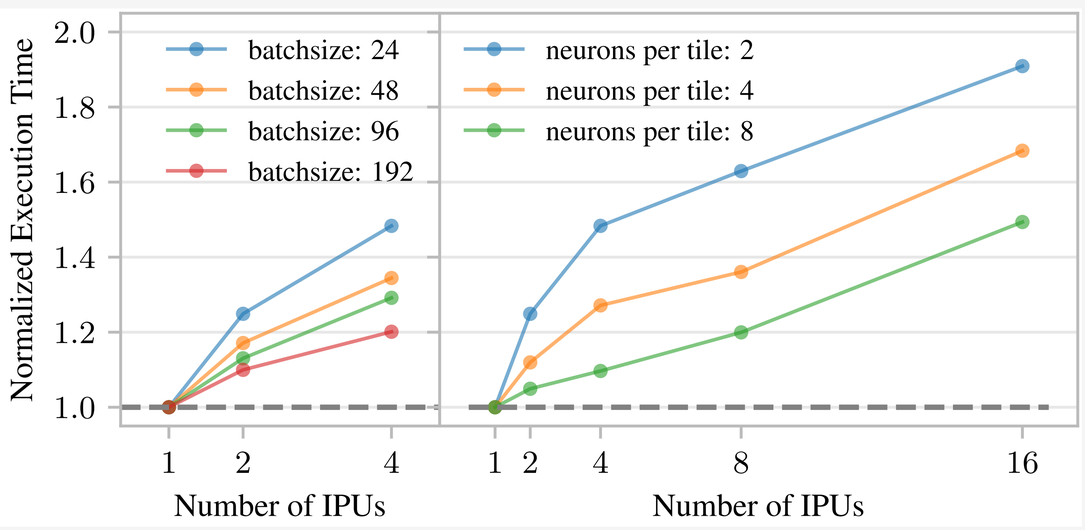
Drastically Improved SNN-Training Throughput on the IPU by Utilizing Activation Sparsity

- We benchmarked the **sparse SNN implementation** for the IPU by evaluating the throughput of multiple SNN models using the SHD [2], N-MNIST [4] and DVSGesture [1] dataset and compared to an equivalent dense implementation on a GPU (NVIDIA GeForce RTX 3090, V100, A100).
- Constant network size with **2944 neurons**.
- Two settings: **"Fixed activity" (top)**, where we force all neurons to spike, approaching the **lower bound in throughput** and **"natural activity" (bottom)** based on random weight initialization, resulting in a conservative approximation of the **upper bound in throughput**.
- The measurement of the acceleration factor includes all necessary operations for the training process, meaning the forward pass, the backward pass to calculate the gradients, and the weight update.
- By utilizing the sparse activations we achieve substantial gains in throughput on the IPU by at least a factor of **5-10× compared to the GPU**.
- For **more extreme levels of activation sparsity** which are still relevant for training runs in practice, even higher acceleration factors of **15-20×** can be achieved.



Great Scalability to Larger Networks and Multi-IPU Settings





Left: Single IPU scaling

- **Larger networks show increasingly higher acceleration** compared to the GPU baseline.
- Increase in network size is achieved by modifying the number of neurons that are allocated on each tile and by extending the number of layers in the network architecture accordingly.

Right: Multi-IPU scaling

- Weak scaling results on SHD dataset with "natural activity" at max activity of 5% up to sizes of 190k neurons and 180 million parameters.
- Shifting from communication-bound towards **compute-bound workloads** by increasing either the batchsize or the network size per IPU **improves the scaling behavior**.

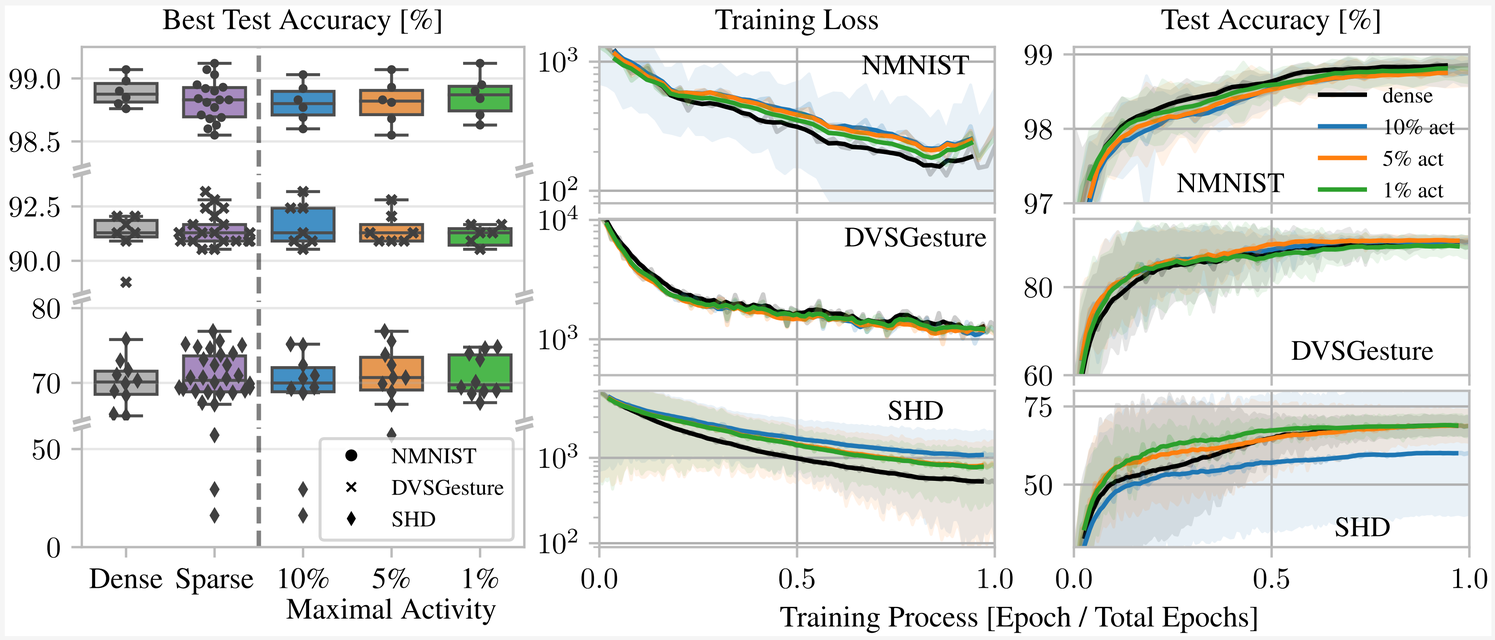
No Slowdown in Training Convergence due to Sparse Training

- In order to train the SNNs we use a backpropagation through time (**BPTT**) algorithm.
- We use a sparse implementation of **surrogate gradient method** [3, 7, 6], where a smooth and differentiable surrogate function is used for the gradient computation in the backward pass. Similar to [5] also propagate information for some neurons that did not spike by introducing a secondary threshold for the gradient computation:

$$S_i^{\text{out}}[t] = \Theta(u_i - \vartheta_i) := \begin{cases} 1, & u_i - \vartheta_i \geq 0 \\ 0, & u_i - \vartheta_i < 0 \end{cases}$$

$$\frac{\partial S_i}{\partial u_i}[t] := \begin{cases} (\beta|u_i - \vartheta_i| + 1)^{-2}, & u_i - \vartheta_{\text{grad}} \geq 0 \\ 0, & u_i - \vartheta_{\text{grad}} < 0 \end{cases}$$

- Using this approach we observe no reduction in neither the final test accuracy (left) nor in the training convergence, meaning the number of epochs required to reach the best test accuracy (right).
  → The demonstrated **increase in training throughput on the IPU directly translates to a reduction in overall training time**.



- Arnon Amir et al. "A Low Power, Fully Event-Based Gesture Recognition System". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.
- Benjamin Cramer et al. "The Heidelberg Spiking Data Sets for the Systematic Evaluation of Spiking Neural Networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 33.7 (2022).
- Emre O Neftci, Hesham Mostafa, and Friedemann Zenke. "Surrogate gradient learning in spiking neural networks: Bringing the power of gradient-based optimization to spiking neural networks". In: *IEEE Signal Processing Magazine* 36.6 (2019).
- Garrick Orchard et al. *Converting Static Image Datasets to Spiking Neuromorphic Datasets Using Saccades*. 2015.
- Nicolas Perez-Nieves and Dan Goodman. "Sparse Spiking Gradient Descent". In: *Advances in Neural Information Processing Systems*. Ed. by M. Ranzato et al. 2021.
- Friedemann Zenke and Surya Ganguli. "SuperSpike: Supervised Learning in Multilayer Spiking Neural Networks". In: *Neural Computation* 30.6 (June 2018).
- Friedemann Zenke and Tim P. Vogels. "The Remarkable Robustness of Surrogate Gradient Learning for Instilling Complex Function in Spiking Neural Networks". In: *Neural Computation* 33.4 (Mar. 2021).