Investigating current-based and gating approaches for accurate and e-efficient spiking recurrent neural networks

Manon Dampfhoffer, Thomas Mesquida, Alexandre Valentian, Lorena Anghel

Investigating Current-Based and Gating Approaches for Accurate and Energy-Efficient Spiking Recurrent Neural Networks

Manon Dampfhoffer[1,2(✉)], Thomas Mesquida[2], Alexandre Valentian[2], and Lorena Anghel[1]

[1] University of Grenoble Alpes, CEA, CNRS, Grenoble INP, INAC-Spintec, 38000 Grenoble, France
manon.dampfhoffer@cea.fr

[2] University of Grenoble Alpes, CEA, List, 38000 Grenoble, France

Abstract. Spiking Neural Networks (SNNs) with spike-based computations and communications may be more energy-efficient than Artificial Neural Networks (ANNs) for embedded applications. However, SNNs have mostly been applied to image processing, although audio applications may better fit their temporal dynamics. We evaluate the accuracy and energy-efficiency of Leaky Integrate-and-Fire (LIF) models on spiking audio datasets compared to ANNs. We demonstrate that, for processing temporal sequences, the Current-based LIF (Cuba-LIF) outperforms the LIF. Moreover, gated recurrent networks have demonstrated superior accuracy than simple recurrent networks for such tasks. Therefore, we introduce SpikGRU, a gated version of the Cuba-LIF. SpikGRU achieves higher accuracy than other recurrent SNNs on the most difficult task studied in this work. The Cuba-LIF and SpikGRU reach state-of-the-art accuracy, only <1.1% below the accuracy of the best ANNs, while showing up to a 49x reduction in the number of operations compared to ANNs, due to the high spike sparsity.

Keywords: SNN · RNN · GRU · Speech recognition

1 Introduction

Artificial Neural Networks (ANNs) have shown impressive results in a wide range of applications such as speech recognition or object detection. However, their energy consumption limits their use in embedded applications. Spiking Neural Networks (SNNs) are a promising research direction targeting the reduction of energy consumption in specialized neuromorphic hardware. SNN computations and communications closely mimick biological neural networks. Spiking neurons communicate with pulses (spikes) instead of continuous-valued activations. They accumulate input spikes in their membrane potential and fire an output spike when the potential reaches a threshold. Similar to biological neural networks,

E. Pimenidis et al. (Eds.): ICANN 2022, LNCS 13531, pp. 359–370, 2022.
https://doi.org/10.1007/978-3-031-15934-3_30

SNNs have an inherent temporal dynamics. They integrate spikes over time and the network inference is performed over several algorithmic timesteps. Therefore, SNNs computations are based on accumulate (AC) instead of multiply-and-accumulate (MAC) operations, which consume more energy [10]. Moreover, SNN computations can be handled in an event-based manner in neuromorphic hardware [7], allowing to exploit their natural spike sparsity.

SNNs have been mostly benchmarked on static vision tasks, such as image classification. However, the inference on static data must be decomposed over several timesteps in order to match the SNN temporal dynamics. Moreover, there is a trade-off between the SNN accuracy and latency (the number of timesteps used to decompose the SNN inference) [8]. On the other hand, SNNs have been less considered for audio applications, although their inherent temporal dynamics may better fit temporal rather than static data. Indeed, the data are already sequential, which means that the latency is not increased compared to a processing by a standard ANN. Moreover, spiking neurons have a self recurrence due to the spike accumulation in the membrane potential over time which may help learning temporal dependencies. Besides, bio-inspired dynamic sensors, such as artificial cochleas [2], are a relevant application for SNNs as they produce data already in the form of spikes. This spiking data can be fed into the SNNs without pre-processing in order to benefit from the high sparsity and high temporal resolution of these sensors [13]. Recently, spiking audio datasets based on a neurophysiology-inspired processing, outputting data in a similar format than dynamic audio sensors, have been proposed to benchmark SNNs [6,16].

SNNs for deep learning applications are based on variants of the Leaky Integrate-and-Fire (LIF) model [1]. For instance, in the Current-based LIF (Cuba-LIF) model, spikes are integrated into a current variable prior to the membrane potential. Moreover, artificial neuron models can be used with recurrent topologies to improve the accuracy on sequential data. In addition, gated recurrent networks, such as the Long Short-Term Memory (LSTM) [9] and the Gated Recurrent Unit (GRU) [5] models, have been proposed to improve the performance of simple Recurrent Neural Networks (RNNs).

In this paper, we investigate the performance of LIF and Cuba-LIF models with recurrent topologies on three spiking audio datasets from a Dynamic Audio Sensor (DASDIGITS [2]) or from a neurophysiology-inspired pre-processing (SHD and SSC [6]), for digits and single words classification. Moreover, we introduce the Spiking Gated Recurrent Unit (SpikGRU), which is an extension of the Cuba-LIF with a gate. Finally, we compare the accuracy and energy-efficiency of the LIF, Cuba-LIF, SpikGRU and ANN models (RNN and GRU). The main contributions of this paper are summarized as follows:

- We show that, for processing temporal sequences, the Cuba-LIF outperforms the LIF model by showing higher accuracy for a similar energy-efficiency.
- We propose SpikGRU, a novel spiking gated recurrent model achieving higher accuracy than other spiking models on the most difficult task (SSC).
- We demonstrate state-of-the-art accuracy compared to previous works using SNNs on the SHD and SSC datasets, bridging the gap with ANN accuracy, while showing up to a 49x improvement in energy compared to the GRU.

2 Related Work

2.1 Leaky Integrate-and-Fire and Current-Based Models

The LIF model is commonly used in SNNs for deep learning applications. The LIF model with a recurrent network topology can be described as:

$$v_t^l = \beta \odot v_{t-1}^l + W_v s_t^{l-1} + U_v s_{t-1}^l + b_v - v_{th} s_{t-1}^l \tag{1}$$

$$s_t^l = H[v_t^l - v_{th}] \tag{2}$$

v_t^l and s_t^l are vectors corresponding respectively to the membrane potential and output spikes of neurons from layer l at time t. $\odot$ denotes element-wise multiplication. Spike firing happens when the membrane potential is superior to the threshold v_{th}, which corresponds to the Heaviside step function H. After each spike, v_{th} is substracted from the membrane potential of spiking neurons. The parameters of the models are W_v and U_v, the weight matrices of feed-forward and recurrent connections (resp.), and b_v, the bias vector. The time constant β corresponds to an exponential decay of v over time.

Neuron models with more temporal dynamics than the simple LIF model can achieve superior accuracy for processing temporal data. For instance, recent works [3,22,23] show the superiority of the Adaptive LIF (Adapt-LIF) over the LIF model for speech recognition. Adapt-LIF uses an adaptive threshold with temporal dynamics (the threshold is increased after each spike fired and decays exponentially with time). In addition, heterogeneous time constant parameters learned per neuron (as opposed to fixed for a layer) can improve the learning on temporal data, allowing the neurons to specialize at different time scales [17]. The Cuba-LIF model is another variant of the LIF introducing an input current i, which integrates the incoming spikes before transmitting them to v with a time constant α and parameters W_i, U_i and b_i. v_t^l is thus defined as a linear combination of its previous state v_{t-1}^l and input i_t^l. Note that in our work, α and β time constants of LIF and Cuba-LIF models are defined as vectors (different constants per neuron) of trainable parameters as in [17]. We use the following definition of the Cuba-LIF model, similar to [17]:

$$i_t^l = \alpha \odot i_{t-1}^l + W_i s_t^{l-1} + U_i s_{t-1}^l + b_i \tag{3}$$

$$v_t^l = \beta \odot v_{t-1}^l + (1 - \beta) \odot i_t^l - v_{th} s_{t-1}^l \tag{4}$$

$$s_t^l = H[v_t^l - v_{th}] \tag{5}$$

2.2 Gated Recurrent Networks

RNNs learn temporal dependencies by reusing the information from previous timesteps due to the recurrent connections. However, their training can be unstable due to vanishing and exploding gradient problems, which can prevent the learning of long-term dependencies [4]. Gated RNNs, such as LSTM and GRU,

can mitigate these problems. Indeed, the gating mechanism allows to better control the flow of information over the timesteps and can create temporal shortcuts which prevent gradient vanishing. Some gated SNNs inspired by the LSTM model have been proposed [14,18,20]. In [20], a LSTM is converted to a spiking version using piece-wise linear counterparts for the activation functions. A spiking LSTM model that can be directly trained with backpropagation through time is proposed in [14]. A hybrid analog and spiking LSTM is demonstrated in [18]. This hybrid network benefits from event-based spike accumulation, but at the expense of decomposing each LSTM timestep into 128 SNN timesteps. However, the LSTM model is computationally expensive due to the use of three gates per unit, which highly increases the number of synaptic operations per layer compared to a simple RNN. The GRU and its variants demonstrate that it is possible to achieve similar accuracy with fewer gates per unit [5,19].

3 SpikGRU: A Spiking Gated Recurrent Unit

We investigate the benefits of gated units in recurrent SNNs by proposing a new model: SpikGRU (Spiking Gated Recurrent Unit). It is inspired by the current-based approach of the Cuba-LIF and the gated approach of the Light-GRU [19], a light version of the GRU model with a single gate. Indeed, SpikGRU can be seen as an extension of the Cuba-LIF model with an additional gate, z, instead of the parameter β. z is computed using the incoming spikes and another set of parameters, W_z, U_z and b_z, and is processed with a sigmoid activation function. The purpose of z is to determine the best combination of the previous state v_{t-1}^l and the input current (or candidate state) i_t^l used in the computation of v_t^l, similar to the update gate in the Light-GRU. We define SpikGRU as:

$$i_t^l = \alpha \odot i_{t-1}^l + W_i s_t^{l-1} + U_i s_{t-1}^l + b_i \tag{6}$$

$$z_t^l = \sigma(W_z s_t^{l-1} + U_z s_{t-1}^l + b_z) \tag{7}$$

$$v_t^l = z_t^l \odot v_{t-1}^l + (1 - z_t^l) \odot i_t^l - v_{th} s_{t-1}^l \tag{8}$$

$$s_t^l = H[v_t^l - v_{th}] \tag{9}$$

Figure 1 illustrates the comparison between the LIF, Cuba-LIF and SpikGRU models. Unlike other spiking versions of gated networks [14,20] we did not use spikes to transmit information between the variables (i, z, v) but instead we transmit directly the value of the variable, which is continuous (represented as a floating point value in our simulations). This is similar to the idea of the Cuba-LIF where v takes i as input, introducing element-wise multiplications instead of only additions. This increases the accuracy (as there is no discretization of the information) at the expense of only a small increase in energy consumption. Indeed, these operations occur only at the neuron level and not at each synapse, the number of synapses being proportional to the square of the number of neurons in a fully connected topology. Moreover, contrary to LSTM networks, we use a single gate instead of three, which limits the computational cost of the model.

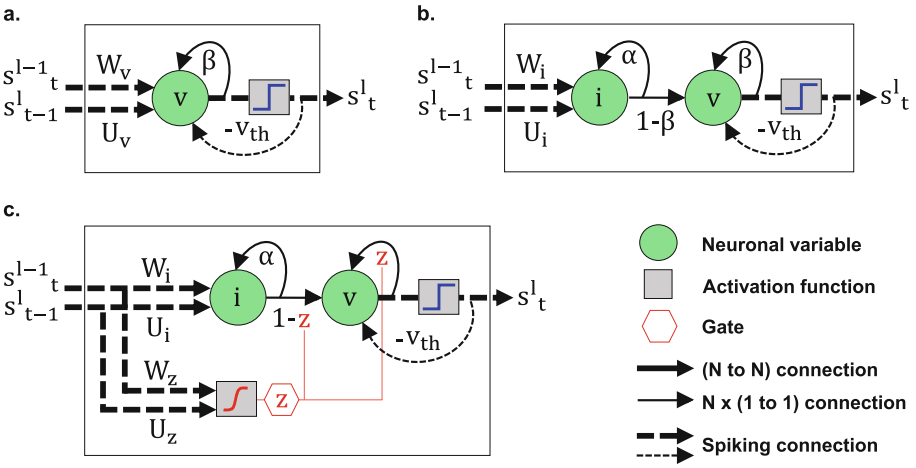

Fig. 1. Recurrent SNN models described in equations (1–9), considering a layer with input and output size N and omitting biases for clarity. a. LIF. b. Cuba-LIF. c. Proposed SpikGRU.

4 Experiments

4.1 Methods

Datasets and Pre-processing. We used three spiking datasets with a classification task to benchmark our SNN models with different degrees of task complexity. DASDIGITS [2] corresponds to the recording from a Dynamic Audio Sensor (64 channels) of the TIDIGITS audio dataset. DASDIGITS consists of 11 classes corresponding to the english digits "one" to "nine" plus "oh" and "zero", spoken by 111 (resp. 109) individuals for training (resp. testing) samples. The single digit version of the dataset contains 2,464 training and 2,486 testing samples. We used the dataset from the CochleaAMS1b sensor and a constant time bin pre-processing 200 Hz. We cut the samples after 1.25 s (almost no spikes are emitted from the sensor after that time) to obtain samples of length 250 timesteps. Therefore, at each timestep, the spike count (number of spikes produced during the time bin) from each channel is fed to both SNN and ANN models in order to compare them with the same data pre-processing. SHD and SSC [6] are created with an audio-to-spiking conversion procedure inspired by neurophysiology using 700 channels. SHD is a spiking version of the Heidelberg Digits audio dataset consisting in 20 classes of spoken digits in English and German from 12 speakers. It contains 8,156 training and 2,264 testing samples. The test set contains samples from 2 individuals that are not used in the training set plus 5% of samples from other speakers. SSC is a more difficult task based on the Google Speech Command dataset. It contains 35 classes corresponding to 35 english words (digits, single word commands and auxiliary words).

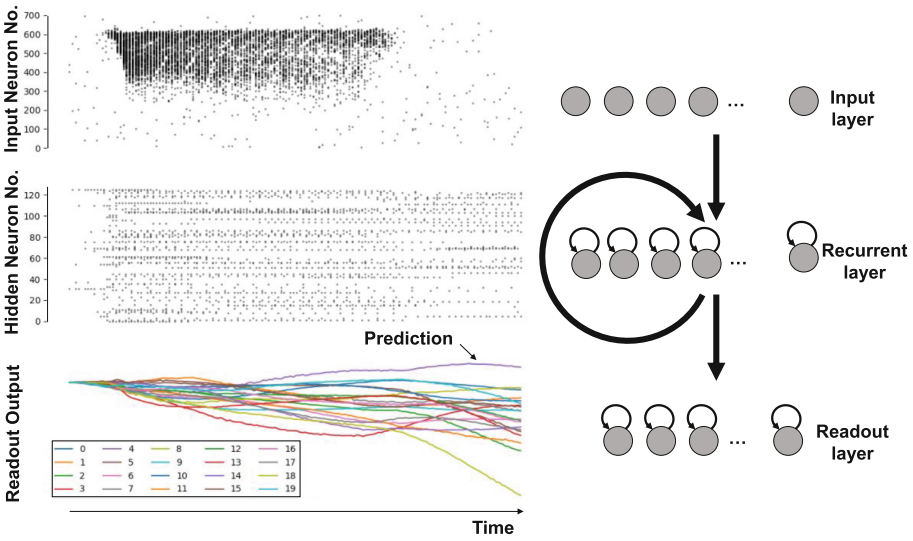

Fig. 2. Sample from the SHD dataset and response from a SNN with 1 recurrent layer.

In this dataset, samples from 1864 individuals are randomly split between training (75,466), validation (9,981) and test (20,382) sets. SHD and SSC samples have 1s duration and spikes are binned 250 Hz. The obtained spike count is also fed directly to the models at each of the 250 timesteps.

Training Procedure. We use network topologies with 1 or 2 recurrent layers of 128 units and a readout layer (fully-connected to the last recurrent layer), as shown in Fig. 2. The readout layer consists of neurons integrating inputs with a self-recurrence, similar to LIF neurons, without the spiking and resetting mechanisms. We use this readout layer for all models as it increases the accuracy compared to a standard fully connected layer, except for the GRU. For the training with DASDIGITS and SHD, we set 20% and 10% (resp.) of the training set as validation set. To avoid overfitting on the SHD and SSC datasets, we introduce noise in the input samples during training using spike jitter across channels, as in [6,17]. We use the max-over-time loss described in [6], which is the cross-entropy loss applied on the maximum value of the neurons of the readout layer over all timesteps. All models are trained with backpropagation through time using Adam [11] optimizer with a learning rate 0.001 for 200 epochs and a batch size 128 (512 for SSC). The standard RNN model leads to unstable training and low accuracy on these tasks. We mitigated theses problems by initializing the recurrent weight matrices with the identity matrix scaled by a factor (0.5) and using the Rectified Linear Unit (ReLU) activation function as proposed in [12]. For the SNNs, weights and biases are initialized from a uniform distribution $U(-k^{-1/2}, k^{-1/2})$, with k being the input size of the layer. The time constants

α and β are learnable parameters per neuron and initialized at 0.9. During training, they are clipped between 0 and 1 to avoid unstable behaviors. The spiking threshold v_{th} is set to 1. The input currents i and membrane potentials v are clipped during training as we observe it improves the accuracy. As the spiking activation function is not differentiable, we define a surrogate gradient using a piece-wise linear triangular function [15].

4.2 Results and Discussion

Table 1 shows the average accuracy of the SNN and ANN models on the three datasets with the 1×128 and 2×128 topologies. We compare our results with previous works on recurrent SNNs on these datasets (except for DASDIGITS for which we are not aware of other works using similar settings). For all three tasks, the GRU achieves the best accuracy, except with the 2-layer topology for SSC and SHD where it is similar to the RNN and Cuba-LIF, respectively. However, these tasks may be too easy for the GRU. Indeed, the accuracy is not significantly increased from the 1-layer to the 2-layer topology for DASDIGITS and SHD compared to spiking models. Moreover, for the SSC task, the GRU shows a high level of overfitting, which is not entirely solved by the addition of spike jitter accross input channels. We observe that the RNN trained with the special settings described in the previous section has similar accuracy than the GRU on the DASDIGITS and SSC tasks. However, this RNN does not reach a satisfactory average accuracy on the SHD task, partly due to an unstable training, as shown by the large confidence interval. It is interesting to note that spiking RNNs (LIF and Cuba-LIF) do not present such training instability. This may be due to the self recurrence of spiking neurons that is weighted by a time constant with value close to (but lower than) 1, which may help preventing gradient vanishing.

Comparing SNN models, we observe that the accuracy of the LIF is below the Cuba-LIF on all tasks, up to a 8.4% difference on the SSC task with the 1-layer topology. The 2-layer Cuba-LIF yields 85.5% accuracy on DASDIGITS, which is <1% below the accuracy of the 1-layer and 2-layer GRU. On SHD, the 2-layer Cuba-LIF achieves 87.8% accuracy, which is superior to the accuracy of the 1-layer and 2-layer GRU (86.8% and 87.3% resp.). For the more difficult SSC task, SpikGRU outperforms other spiking models for both topologies. Indeed, SpikGRU achieves 74.7% (resp. 77.0%) accuracy with 1-layer (resp. 2-layer) topology, which is only 0.8% (resp. 1.1%) below the best ANN accuracy. Moreover, all our spiking models show higher accuracy on the SHD task than the Adapt-LIF in [22], for the same topology and number of timesteps. However, they use strictly binary inputs, meaning that if there is more than one spike in the time bin it is considered as 1. On the other hand, we directly used the spike count. Indeed, the average input sparsity measured on the testset is only increased from 4.6% to 4.7% (resp. 4.7% to 4.8%) spikes per neuron per timestep on SHD (resp. SSC) for a pre-processing 250 Hz. Therefore, the additional energy consumption is small while the model accuracy is increased as no spikes are lost. Note that, on SHD and SSC, for a pre-processing with high frequency (such as

Table 1. Testing accuracy (%) of the spiking (LIF, Cuba-LIF, SpikGRU) and non-spiking (RNN, GRU) models on the DASDIGITS, SHD and SSC datasets, shown with the 95% confidence interval. The best accuracy for each topology for spiking and non-spiking models is highlighted. Results from related works are also indicated.

	DASDIGITS	SHD	SSC
1×128 network			
GRU	**85.9** ± 1.4	**86.8** ± 1.2	**75.5** ± 0.2
RNN	85.8 ± 1.4	74.9 ± 3.1	75.3 ± 0.7
LIF	78.3 ± 1.9	80.6 ± 2.0	63.1 ± 0.8
Cuba-LIF	81.1 ± 1.1	**83.7** ± 1.3	71.5 ± 0.4
SpikGRU	**81.8** ± 1.1	**83.7** ± 1.5	**74.7**± 0.4
*Adapt-LIF** [22]	–	79.4	–
Cuba-LIF† [6]	–	71.4	50.9
Cuba-LIF† [17]	–	82.7	60.1
2×128 network			
GRU	**86.2** ± 1.3	**87.3** ± 0.9	77.9 ± 0.3
RNN	84.9 ± 1.4	75.0 ± 7.3	**78.1** ± 0.3
LIF	82.7 ± 0.8	85.8 ± 1.7	70.3 ± 1.3
Cuba-LIF	**85.5** ± 0.9	**87.8** ± 1.1	75.7 ± 0.2
SpikGRU	83.3 ± 1.7	86.4 ± 1.8	**77.0** ± 0.4
*Adapt-LIF** [22]	–	84.4	–
Adapt-LIF [23]	–	87.8	74.2‡

* Binary inputs. † 2000 Hz pre-processing. ‡ 2×400 network.

2000 Hz Hz), spike count and binary inputs are equivalent as there is never more than one spike per time bin. Our Cuba-LIF also achieves better accuracy than the Cuba-LIF from [6,17] on both the SHD and SSC datasets for the same topology. However, in [6,17], the pre-processing is set at 2000 Hz Hz which results in 2000 timesteps. The higher the number of timesteps, the higher the precision of the inputs, but also the higher the difficulty of the task. Indeed, it increases the sequence length, making it harder for recurrent units to retain relevant information. The lower accuracy of the Cuba-LIF in [6] can be explained by the fact that they use fixed time constants per layer [17]. The best results among the previous works with SNNs on SHD and SSC datasets are demonstrated in [23], also using 250 Hz pre-processing. For the same topology their Adapt-LIF network shows the same accuracy (87.8%) as our Cuba-LIF on SHD. However, in the SSC task, even with a larger topology (2×400), the accuracy of their Adapt-LIF (74.2%) is lower than the accuracy of our 2-layer Cuba-LIF (75.7%) and SpikGRU (77.0%).

5 Energy-Efficiency

In this section, we compare the energy-efficiency of the previously presented models based on the total effective number of MAC and AC operations. We did

not translate the MAC and AC operations into their respective energy consumption because most of the energy consumption of neural networks in specialized architectures comes from memory accesses associated with arithmetic operations rather than from the arithmetic operations themselves [10]. However, memory accesses cannot be predicted only based on the number of arithmetic operations, as they also depends on data reuse and sparsity exploitation, which are highly architecture-dependent [21]. Therefore, in the interests of comparing the different neuron models, we have ignored the energy associated with memory accesses, and have used the number of MAC and AC operations as a figure of merit for energy efficiency.

Spiking models exhibit a high sparsity. On the given tasks, our spiking models produce on average between 0.06 and 0.21 spikes per neuron per timestep for processing one sample. The 2-layer Cuba-LIF yields 0.06 spikes per neuron per timestep on DASDIGITS and SSC, which means that a neuron produces on average only 15 spikes during the 250 timesteps (or 1 spike every 17 timesteps). Similarly, the 2-layer SpikGRU achieves 0.09 spikes per neuron per timestep on SSC. Therefore, the number of operations per sample is highly reduced compared to an ANN where operations are performed at each timestep. Table 2 indicates the number of MAC and AC operations per timestep of one layer of the ANN and SNN models to process a sample. We observe that in ANN models (GRU and RNN) there are mainly MAC operations (except for the bias of neurons), while in SNN there are mainly AC operations (and some element-wise multiplications). In SNN models, the number of AC is weighted by the activity rate (spikes per neuron per timestep) of the SNN layers, which decreases (resp. increases) the number of operations if it is inferior (resp. superior) to 1, compared to an ANN. Note that the Cuba-LIF has similar number of operations than the LIF. Indeed, the input current variable represents only additional MACs at the neuron level, which is negligible compared to the operations in the feedforward and recurrent synaptic connections. On the other hand, the SpikGRU model increases significantly the number of operations compared to LIF and Cuba-LIF due to the additional feedforward and recurrent synaptic connections.

Figure 3 shows the accuracy vs. total effective number of operations (MAC + AC) per timestep of SNN and ANN models on the three datasets. The number of operations in the 2-layer Cuba-LIF is decreased by 16x compared to the 1-layer GRU while the models have similar accuracy on DASDIGITS. On SHD, the 2-layer Cuba-LIF even slightly outperforms the 1-layer and 2-layer GRU while reducing by 37x and 49x (resp.) the number of operations. On SSC, the number of operations in the 2-layer SpikGRU is reduced by 8x (resp. 24x) while the model yields an accuracy only ≈1% below the accuracy of the 2-layer RNN (resp. GRU). Compared to the Cuba-LIF on SSC, the SpikGRU model shows better accuracy but at the expense of 2x the number of operations. Our models are compared with the Adapt-LIF from [23] using the number of MAC and AC operations provided in their paper. Our most accurate 2-layer spiking models are more energy-efficient than the Adapt-LIF. Indeed, the number of operations per timestep is 8.6k (Cuba-LIF) vs. 11.5k for the SHD task, and 17.6k (SpikGRU) vs. 28.5k for SSC.

Table 2. Number of MAC and AC operations per timestep per sample for one layer of the ANN and SNN models. m and n are respectively input and output size of the layer. For SNN models, a_{in} and a_{out} are respectively input and output activity rate (spikes per neuron per timestep) of the layer.

Model	Nb MAC	Nb AC
GRU	$3mn + 3n^2 + 3n$	$3n$
RNN	$mn + n^2$	n
LIF	n	$mn * a_{in} + (n^2 + n) * a_{out} + n$
Cuba-LIF	$3n$	$mn * a_{in} + (n^2 + n) * a_{out} + n$
SpikGRU	$3n$	$2mn * a_{in} + (2n^2 + n) * a_{out} + 2n$

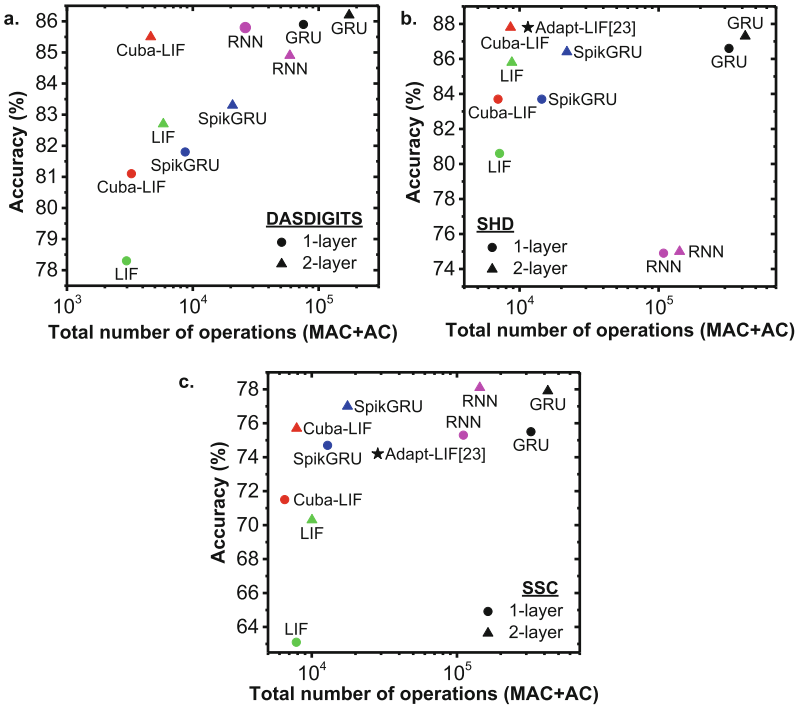

Fig. 3. Accuracy vs. total number of operations (MAC + AC) per timestep for processing one sample from the (a) DASDIGITS, (b) SHD and (c) SSC datasets.

6 Conclusion and Perspectives

Our experiments on the DASDIGITS, SHD and SSC datasets demonstrate the ability of recurrent SNNs to perform classification on sequential data with high energy-efficiency. The number of operations in the Cuba-LIF and proposed SpikGRU models is reduced by up to 49x and 24x (resp.) compared to the GRU, for

almost the same accuracy (<1.1% below). Moreover, we demonstrate that the Cuba-LIF model outperforms the LIF model, as it achieves better accuracy for approximately the same number of operations. In addition, the Cuba-LIF may also outperform the Adapt-LIF model for these tasks. Indeed, the Cuba-LIF achieved better accuracy than the Adapt-LIF from previous works, for a similar model complexity. Moreover, our proposed SpikGRU model shows a high potential to outperform non-gated recurrent SNNs on more difficult tasks, at the expense of an increased number of operations. However, this must be further investigated. Indeed, we studied tasks with different degrees of difficulty, due to the input size and number of classes, but we must also evaluate its ability to retain longer-term dependencies than the Cuba-LIF using tasks with different temporal sequence length. Besides, our results show that the number of operations in SNNs is highly dependent on their spiking activity. However, in our work, we did not specifically tune the spiking activity of the SNN models. Therefore, methods to boost sparsity in SNNs will result in further energy savings.

Acknowledgements. This work has been partially supported by MIAI @ Grenoble Alpes, (ANR-19-P3IA-0003).

References

1. Abbott, L.: Lapicque's introduction of the integrate-and-fire model neuron (1907). Brain Res. Bullet. **50**(5), 303–304 (1999). https://doi.org/10.1016/S0361-9230(99)00161-6
2. Anumula, J., Neil, D., Delbruck, T., Liu, S.C.: Feature representations for neuromorphic audio spike streams. Front. Neurosci. **12** (2018). https://doi.org/10.3389/fnins.2018.00023
3. Bellec, G., Salaj, D., Subramoney, A., Legenstein, R.A., Maass, W.: Long short-term memory and learning-to-learn in networks of spiking neurons. In: Advances in Neural Information Processing Systems: NeurIPS, pp. 795–805 (2018)
4. Bengio, Y., Simard, P., Frasconi, P.: Learning long-term dependencies with gradient descent is difficult. IEEE Trans. Neural Netw. **5**(2), 157–166 (1994). https://doi.org/10.1109/72.279181
5. Cho, K., et al.: Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1724–1734. Association for Computational Linguistics, Doha (2014). https://doi.org/10.3115/v1/D14-1179
6. Cramer, B., Stradmann, Y., Schemmel, J., Zenke, F.: The Heidelberg spiking data sets for the systematic evaluation of spiking neural networks. IEEE Trans. Neural Netw. Learn. Syst. 1–14 (2020). https://doi.org/10.1109/TNNLS.2020.3044364
7. Davies, M., et al.: Loihi: a neuromorphic manycore processor with on-chip learning. IEEE Micro **38**(1), 82–99 (2018). https://doi.org/10.1109/MM.2018.112130359
8. Han, B., Srinivasan, G., Roy, K.: RMP-SNN: residual membrane potential neuron for enabling deeper high-accuracy and low-latency spiking neural network. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13555–13564 (2020). https://doi.org/10.1109/CVPR42600.2020.01357

9. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
10. Horowitz, M.: Computing's energy problem (and what we can do about it). In: 2014 IEEE International Solid-State Circuits Conference Digest of Technical Papers (ISSCC), pp. 10–14 (2014). https://doi.org/10.1109/ISSCC.2014.6757323
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arxiv:1412.6980 (2014)
12. Le, Q.V., Jaitly, N., Hinton, G.E.: A simple way to initialize recurrent networks of rectified linear units. arXiv preprint arXiv:1504.00941 (2015)
13. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128× 128 120 db 15 μs latency asynchronous temporal contrast vision sensor. IEEE J. Solid-State Circuits **43**(2), 566–576 (2008). https://doi.org/10.1109/JSSC.2007.914337
14. Lotfi Rezaabad, A., Vishwanath, S.: Long short-term memory spiking networks and their applications. In: International Conference on Neuromorphic Systems 2020, pp. 1–9. ACM (2020). https://doi.org/10.1145/3407197.3407211
15. Neftci, E., Mostafa, H., Zenke, F.: Surrogate gradient learning in spiking neural networks: bringing the power of gradient-based optimization to spiking neural networks. IEEE Signal Process. Magaz. **36**, 51–63 (2019). https://doi.org/10.1109/MSP.2019.2931595
16. Pan, Z., Chua, Y., Wu, J., Zhang, M., Li, H., Ambikairajah, E.: An efficient and perceptually motivated auditory neural encoding and decoding algorithm for spiking neural networks. Front. Neurosci. **13** (2020). https://doi.org/10.3389/fnins.2019.01420
17. Perez-Nieves, N., Leung, V.C.H., Dragotti, P.L., Goodman, D.F.M.: Neural heterogeneity promotes robust learning. Nature Commun. **12**(1), 5791 (2021). https://doi.org/10.1038/s41467-021-26022-3
18. Ponghiran, W., Roy, K.: Hybrid analog-spiking long short-term memory for energy efficient computing on edge devices. In: 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 581–586 (2021). https://doi.org/10.23919/DATE51398.2021.9473953
19. Ravanelli, M., Brakel, P., Omologo, M., Bengio, Y.: Light gated recurrent units for speech recognition. IEEE Trans. Emerg. Topics Comput. Intell. **2**(2), 92–102 (2018). https://doi.org/10.1109/TETCI.2017.2762739
20. Shrestha, A., et al.: A spike-based long short-term memory on a neurosynaptic processor. In: 2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 631–637 (2017). https://doi.org/10.1109/ICCAD.2017.8203836
21. Sze, V., Chen, Y.H., Yang, T.J., Emer, J.S.: Efficient processing of deep neural networks: a tutorial and survey. Proc. IEEE **105**(12), 2295–2329 (2017). https://doi.org/10.1109/JPROC.2017.2761740
22. Yin, B., Corradi, F., Bohté, S.M.: Effective and efficient computation with multiple-timescale spiking recurrent neural networks. In: International Conference on Neuromorphic Systems 2020, pp. 1–8. ACM (2020). https://doi.org/10.1145/3407197.3407225
23. Yin, B., Corradi, F., Bohté, S.M.: Accurate and efficient time-domain classification with adaptive spiking recurrent neural networks. Nat. Mach. Intell. **3**(10), 905–913 (2021). https://doi.org/10.1038/s42256-021-00397-w