PositCL: Compact Continual Learning with Posit Aware Quantization

Vedant Karia, Abdullah M. Zyarah, and Dhireesha Kudithipudi
Neuromorphic AI Lab, University of Texas at San Antonio
San Antonio, TX, USA

ABSTRACT

Neural network models catastrophically forget previously learned information while acquiring new knowledge, requiring a fundamental change in learning models and architectures. These enhancements to architecture structures and training mechanisms lead to an increase in memory and computational resources, making it difficult to deploy models on resource-constrained edge devices. To enhance both memory and computational efficiency, we propose a model compression approach for spiking continual learning models, where the model parameters are quantized with varying precision according to their weight distribution.

Specifically, we explore the posit format with gradient scaling and gradient accumulation techniques to reduce the quantization error of the model while training. Synapses and regularization parameters that play a role in catastrophic forgetting are designed with an 8-bit posit format. The model exhibits a 4× reduction in memory with a marginal impact ≈ 2% on mean accuracy. This model also exhibits a 30% increase in mean accuracy compared to the 8-bit fixed point. We show that the posit spiking network consumes 27% less energy compared to the 16-bit fixed point for similar performance.

KEYWORDS

Spiking neural networks, Continual learning, Low-precision arithmetic, Posit numerical format.

ACM Reference Format:
Vedant Karia, Abdullah M. Zyarah, and Dhireesha Kudithipudi. 2024. PositCL: Compact Continual Learning with Posit Aware Quantization. In *Great Lakes Symposium on VLSI 2024 (GLSVLSI '24), June 12–14, 2024, Clearwater, FL, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3649476.3660371

1 INTRODUCTION

Neural network models have shown promising performance on non-overlapping tasks with static underlying patterns. In contrast, when trained sequentially on multiple tasks, they tend to forget the knowledge from the previous tasks, a problem referred to as catastrophic forgetting. This problem can be framed as a stability-plasticity dilemma, with stability representing the preservation of prior knowledge and plasticity indicating the ability to learn new knowledge. Several continual learning (CL) mechanisms have been proposed in the literature [3, 18, 28] to achieve the stability-plasticity balance during the learning process. Common methods to address this issue include parameter regularization [1], replay or rehearsal techniques [16], and dynamic architectures [24].

Several works have incorporated such mechanisms into neural network models to improve continual learning capabilities at different granularities [25]. However, a common issue among all these approaches is that they are compute and memory intensive [3, 11], making them unsuitable for resource-constrained edge devices. Previous studies explored regularization methods in spiking neural networks (SNNs) to reduce compute and memory resources [23]. SNNs enable event-driven computation and efficient data encoding and processing, where information is represented in the form of sparse binary spike streams instead of high-precision data. This serves two purposes: i) it allows short- and long-term information retention [20], ii) it can lead to a reduction in computational cost and power consumption by several orders of magnitude [8, 21].

There are limited explorations of optimizing continual learning models during inference and learning, even more so for spiking networks. The loss of information during the compression process, which involves reducing the precision of the model parameters in SNNs, has led to catastrophic degradation of the network performance [19]. Existing SNN model compression techniques focus on quantizing the model parameters for the inference phase to shorten the latency and enhance energy efficiency. SNN quantization has also been shown to reduce memory footprint by ≈ 4× while maintaining accuracy within marginal degradation compared to baseline [19].

It should be mentioned that there are several compression techniques that one can incorporate in SNN models, such as pruning, low-precision quantization, and low-rank factorization. Pruning and quantization techniques focus mainly on reducing the redundancy in the model, while low-rank factorization uses matrix/tensor decomposition to identify the key parameters. In this work, we will leverage the robustness of neural networks for quantization [6] and emphasize on low-precision representation. Specifically, we study: i) quantization of the parameters to tapered-low precision [13, 14] which outperforms other approaches, ii) integration of posit-based quantization techniques into continual learning models with acute awareness of computational and memory constraints inherent to edge devices.

The main contributions of this paper are:

(1) A posit-quantized continual learning mechanism to attenuate the effects of catastrophic forgetting that is compute- and memory-efficient.

GLSVLSI '24, June 12–14, 2024, Clearwater, FL, USA

ACM ISBN 979-8-4007-0605-9/24/06
https://doi.org/10.1145/3649476.3660371

(2) We examine energy dissipation using an analytical approach for the continual learning network quantized to 8-bit posit and 16-/8-bit fixed-point representations.

2 BACKGROUND

2.1 Continual Learning

Neural network models often face the issue of catastrophic forgetting when learning multiple tasks. Previous studies address catastrophic forgetting by minimizing the overlap in representation between tasks [4, 5], replaying samples from previously learned tasks [2, 22], or penalizing changes to critical parameters (regularization) to safeguard previously acquired knowledge from interference. Our exploration in this work will be focused on the regularization method. One notable data-focused regularization method, Learning without Forgetting (LwF) [15], utilizes previous task models as soft labels for earlier tasks. Other approaches, such as Elastic Weight Consolidation (EWC) [10] and Synaptic Intelligence (SI) [26], estimate the importance of network parameters and penalize changes to crucial parameters during subsequent task training. SI further extends the EWC to use adaptable regularization. Despite the effectiveness of the aforementioned approaches in addressing catastrophic forgetting, they often triple the memory requirements compared to base models. To address this challenge of memory, TACOS [23] introduces metaplasticity combined with synaptic consolidation techniques in a spiking neural network. This approach reduces memory requirements while improving the performance of continual learning.

2.2 Quantization

Quantization is defined as mapping a large set of values to a finite or smaller set of values [6], which is used to approximate the calculations of integrals. In neural networks, which are typically dominated by computationally intensive operations, quantization plays a critical role in mitigating the memory and computational overhead by reducing bit-precision of the network activations and their parameters. Some of the common quantization techniques use uniform quantization [6], where high-precision floating point numbers are quantized to low-precision values as illustrated in Equation 1. Here, x is the value to be quantized, S is the scaling factor, and Z is a constant to achieve symmetric distribution.

$$Q(x) = Int(\frac{x}{S}) + Z \tag{1}$$

Although low-precision uniform quantization has been proven to be effective in reducing memory and computational cost, they can cause a significant drop in network performance [6]. To address this challenge, various non-uniform quantization techniques have been introduced such as posit-based quantization [12, 13] and tapered fixed-point quantization [14] which aligns the parameter distribution with the numerical format distribution to reduce the quantization error. These non-uniform numerical formats outperform the integer and floating point due to their high dynamic range and high precision of the values close to zero [13]. It is important to mention that there are numerous quantization techniques that have been proposed in literature to compress network models without compromising their accuracy while performing inference. However, most of these techniques fail to achieve satisfactory performance during quantized training due to accumulation of quantization error. This problem may escalate further when dealing with continual learning scenarios. Previous works in [9] and [27] targeted quantization for continual learning. The former uses dual-fixed point quantized metaplastic synapses while the latter uses probabilistic metaplastic binary synapses to address catastrophic forgetting.

2.3 Low-precision Posit

The posit numerical format was first introduced by [7] as an alternative representation to the IEEE floating point formats. Due to its higher dynamic range and high resolution compared to the IEEE floats, it was employed at low precision for various applications to avail the memory and computational benefits. Unlike IEEE floats, numbers in posit are represented by Equation (2), where s, es, fs, represents the sign, and the maximum number of bits allocated for the exponent and the maximum value that can be attained by fraction bits. e and f denote the exponent and fraction values, respectively, and k is the the regime value, given by Equation 3.

$$x = \begin{cases} 0, & \text{if } (00...0) \\ NaR, & \text{if } (10...0) \\ (-1)^s \times 2^{2^{es} \times k} \times 2^e \times \left(1 + \frac{f}{2^{fs}}\right), & \text{otherwise} \end{cases} \tag{2}$$

The regime bit-field is encoded based on the *runlength* (m) of identical bits ($r...r$) terminated by either a *regime terminating bit* ($\overline{r}$) or the end of the n-bit value. Note that there is no requirement to distinguish between negative and positive zero since only a single bit pattern ($00...0$) represents zero. Furthermore, instead of defining a *NaN* for exceptional values and infinity by different bit patterns, a single bit pattern ($10...0$), "Not-a-Real" (NaR), represents exception values and infinity. More details about the posit number format can be found in [7].

$$k = \begin{cases} -m, & \text{if } r = 0 \\ m - 1, & \text{if } r = 1 \end{cases} \tag{3}$$

3 POSIT QUANTIZED CONTINUAL LEARNING

In this work, the TACOS [23] spiking continual learning algorithm is used. The TACOS algorithm incorporates multiple local learning mechanisms, such as metaplasticity and synaptic consolidation, to preserve previous knowledge and learn continually, while addressing catastrophic forgetting. It is trained using the surrogate gradient learning rule known as event-driven random back propagation (eRBP) [17].

The main purpose of choosing the TACOS algorithm is two-fold. Firstly, it demonstrates state-of-the-art performance on several continual learning benchmarks. Second, it uses local learning with spiking neurons to improve energy efficiency. The network consists of neuronal units modeled by Leaky Integrate and Fire (LIF) neurons, as described by Equation 4. In the LIF neuron, the synaptic current $I(t)$ is derived from the weighted summation of spikes over time, subsequently influencing the membrane potential. Given that the synapses w_j are encoded in 8-bit posit format, the computation of the membrane potential presents two viable strategies. First, one can preserve the synapses in posit format and decode them to full precision during computations. Alternatively, not only the synapses but also the membrane potential and other relevant variables to

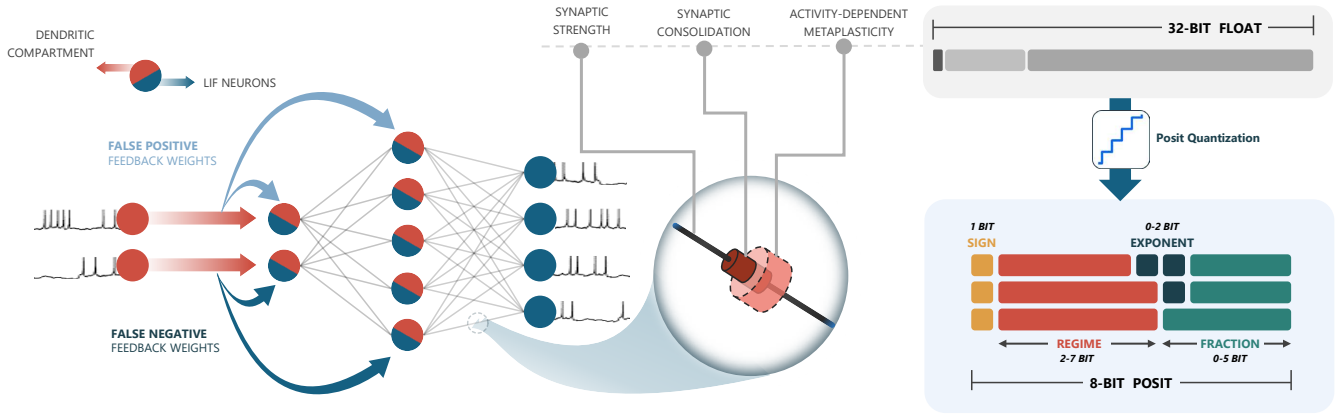

Figure 1: Continual learning spiking neural network architecture with posit quantized synapse parameters, synaptic consolidation parameters, and metaplasticity parameters. The network learns in a single epoch, with 8-bit posit showing only ≈ 3% degradation in mean accuracy compared to 32-bit floating point.

posit format can be represented in posit format. In our approach, we have opted for the former method. Thus, we develop a posit decoder and encoder that converts the 8-bit posit realized synapses to 16-bit fixed point and vice versa. This design choice is derived to reduce the computational complexity of the network while lowering the quantization error of the LIF neuron.

$$V(t+1) = V(t) + \frac{\Delta t}{\tau_{mem}}\left[\left(V_{rest} - V(t)\right) + I(t)R\right] \tag{4}$$

$$I(t+1) = I(t) + \frac{\Delta t}{\tau_{syn}}\left(\sum_{j=1}^{N} w_j S_j(t) - I(t)\right) \tag{5}$$

To mitigate the impact of catastrophic forgetting, the training process for synapses incorporates two crucial parameters linked to each synapse, as depicted in Figure 1: metaplasticity parameters and reference weights. These elements play a critical role in modulating synaptic plasticity and maintaining synaptic strength over long periods, respectively. However, storing each parameter in the network using a 32-bit floating point format imposes significant computational and energy burdens associated with memory storage. To address these resource constraints, our approach quantizes all three parameters related to these mechanisms to 8-bit posit and fixed-point formats. During training, the synapse parameters and reference weights typically converge to values within the range $[1 - 10^{-5}]$, which requires exceptionally high precision close to zero. However, quantizing these parameters to an 8-bit fixed-point representation often results in a majority of values being rounded to zero, thus introducing a considerable quantization error.

To alleviate this error, we adopt 8-bit posit quantization for these parameters, which offers both a high dynamic range and enhanced precision near zero. However, the metaplasticity parameters are determined by accumulating a constant based on the activity of the post-synaptic neuron. This accumulation process results in a linear trajectory for the distribution of these parameters, which aligns well with the uniformly distributed nature of fixed-point representations.

Algorithm 1: PositCL training procedure

Input : Input tasks $\mathcal{T}$, where $T^t \subset \mathcal{T}$ is a set of inputs and target pairs $\{\mathcal{X}^t, \mathcal{Y}^t\}$

for t **in** $\mathcal{T}$ **do**
  for $epoch = 0$ **to** $maxE$ **do**
    for $\{x^t, y^t\}$ **in** $\{X^t, Y^t\}$ **do**
      for τ **in** τ_{sim} **do**
        Network Prediction: $\hat{y}^t = f(x^t)$
        Error Accumulation: $\tau_u \frac{\partial U}{\partial t} = -U + \mathbb{E}R_U$
        Update Neuron Trace: $\frac{d}{dt}X^{tr} = -\frac{X^{tr}}{\tau_{tr}} + S_i(t)$
        // Update Weights:
        for j **in** $S_j(t)$ **do**
          for i **in** $I_{min} \leq I_i \leq I_{max}$ **do**
            $w_{i,j}(t+1) = Quant[w_{i,j}(t) - \eta S_j(t) U_i(t) \Theta(I_i(t)) f(m,w)]$
          end
        end
      end
    end
    for i **in** $S_i(t)$ **do**
      $w_{i,j}(t+1) = Quant[w_{i,j}(t) - \eta_1 f(m,w)(w_{i,j} - w_{i,j}^{ref})]$
    end
    $w_{ij}^{ref} = Quant[w_{ij}^{ref} + \frac{\Delta t}{\tau_{ref}}(w_{ij} - w_{ij}^{ref})]$
    if $x_i^{tr} > m_{th} \& x_j^{tr} > m_{th1}$ **then**
      $m = Quant(m \mathrel{+}= \Delta m)$
    end
  end
end

Therefore, we quantize the metaplasticity parameters into 8-bit fixed-point format.

Table 1: The individual and mean task accuracy for synapse representation with various bit-precision on the Split-MNIST task after training sequentially for a single epoch in the domain-IL scenario. The accuracies in the "Without CL" column reflects the performance of the spiking network without metaplasticity and synaptic consolidation mechanisms at full precision weights.

Task	Without CL	32-bit FP TACOS	16-bits FXP	8-bit FXP	8-bit posit ES=0	8-bit posit ES=1	8-bit posit ES=2
Class 0,1 Acc.	31.54%	**93.84%**	91.76%	46.35%	89.63%	87.78%	88.03%
Class 2,3 Acc.	58.72%	**78.78%**	74.04%	50.80%	70.96%	72.58%	73.18%
Class 4,5 Acc.	13.18%	**69.45%**	61.26%	52.53%	59.15%	59.64%	59.53%
Class 6,7 Acc.	89.93%	92.82.0%	90.87%	87.65%	93.44%	93.39%	**94.30%**
Class 8,9 Acc.	97.73%	77.60%	81.87%	49.01%	**84.37%**	81.97%	81.19%
Mean Accuracy	58.22%	**82.11%**	79.31%	49.37%	79.51%	79.07%	79.30%

$$f(m, w) = e^{-|mw|} \tag{6}$$

Training the network with low-precision parameters introduces a new challenge, where the gradients during the synapse update quantize to zero, effectively freezing the network, thereby hindering further learning and adaptation. To avoid gradient quantization to zero, we adopt the gradient accumulation strategy, where gradients are accumulated at higher precision on a batch of samples before updating the synapses. By accumulating gradients, we increase their range, reducing the likelihood of them quantizing to zero. In addition to gradient accumulation, we use a gradient scaling strategy to further mitigate quantization errors. This strategy involves mapping both synapses and gradients to higher ranges of values suitable for quantization, minimizing the impact of quantization-induced errors. However, gradient scaling leads to computational overhead, as it involves multiplying gradient updates to scale synapses to higher ranges and dividing the quantized parameters to revert them to their original range. Although gradient scaling introduces computational complexity, it plays a crucial role in enhancing the performance of the quantized network by reducing quantization errors and preserving essential information for learning and adaptation.

4 RESULTS AND ANALYSIS

4.1 Continual learning performance

We evaluated the performance of the proposed network on Split-MNIST dataset in a continual learning setting. The dataset splits the MNIST data into five tasks, each containing two classes presented to the network in a sequential fashion. The experiments were performed on 10000 training images and 2000 test images with each task containing 2000 training and 400 test images. The evaluation was performed according to the domain-incremental setting, where the task identity is unknown to the network, and the output neurons are also shared between the tasks. Table 1 illustrates the performance of the network after training on five tasks sequentially with the network topology of 784 input neurons, 200 hidden neurons, and 2 output neurons.

The continual learning performance is calculated using the mean accuracy metric, the average test accuracy across all tasks after performing task-based training. We considered various scenarios and setups to illustrate the impact of regularization techniques (metaplasticity and synaptic consolidation) and quantization on network performance. The baseline network has no regularization techniques incorporated into it, and all synaptic weights are realized using the 32-bit floating-point format. In this setup, we observe that the network tends to forget previously learned tasks when learning new ones. When the network is integrated with metaplasticity and synaptic consolidation mechanisms, it preserves the old knowledge from previous tasks and shows an improvement in the mean accuracy by 24.6% (see 32-bit FP TACOS column). Then, under the same setup, the network parameters were quantized to 16-bit fixed-point and 8-bit fixed-point, and this leads to a significant drop in accuracy as it can be observed in Table 1. In contrast, when the network was quantized to an 8-bit posit numerical format with three variations of exponent bit ranging from 0 to 2, we noticed an enhancement in the mean accuracy compared to 8-bit fixed point and also a balance in stability and plasticity of the model. The increase in mean accuracy can be attributed to two reasons: Firstly, the posit format has the ability to represent the gradients with high-dynamic range and high resolution, unlike the fixed-point representation, which effectively minimizes the quantization error. Secondly, the posit format has the capability to represent small-scale values. In this work, the training is performed in an online fashion and the lack of gradient accumulation set it to zero when quantized with 8-bit fixed point. This eventually freezes the network and halts the learning process.

4.2 Energy analysis

To evaluate the efficiency of the posit quantization technique in continual learning scenarios, we estimate the energy consumption of deploying the quantized system on an edge device. Given the network's complexity, we employ an analytical approach to estimate energy consumption, which implies characterizing the workload of the individual computation units involved in network training and then estimating the energy cost. Notably, computational units such as accumulators, leaky integrate-and-fire neurons, exponential functions, and multipliers are designed and synthesized under the IBM 65nm technology node using the Synopsys Design Compiler (DC) tool, and their energy profile is achieved using the Synopsys PrimeTime-PX tool, as recorded in Table 2. Additionally, energy requirements for reading from and writing to 16-bit and 8-bit memory are determined using the HP Cacti tool. It is crucial to mention here that to compute LIF neuron activations and synapse updates, the posit-quantized synapses need to be encoded into a 16-bit fixed point and then decoded back into posit format after computation. This necessitates the implementation of posit encoder and decoder modules. These critical components were developed with insights drawn from the approach outlined in [12].

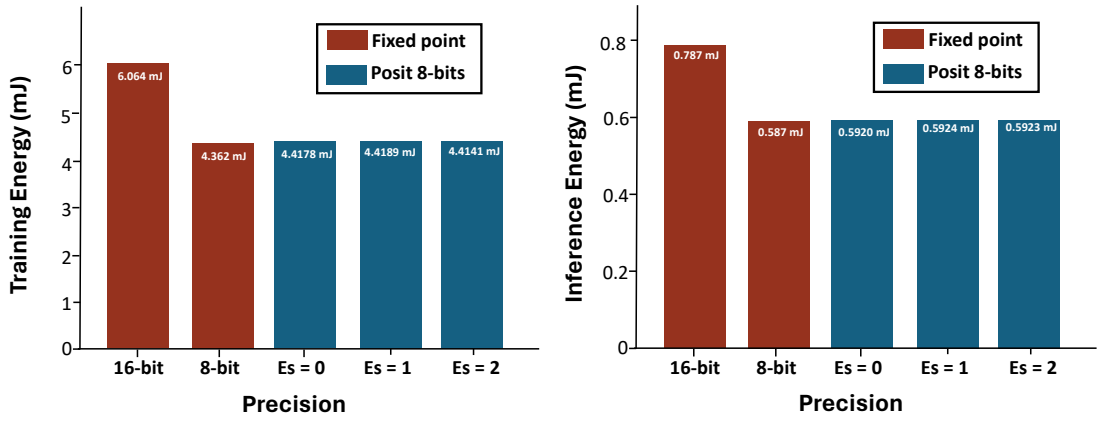

Figure 2: (a) Comparison of the training energy of the network with 16-bit fixed point and 8-bit posit numerical format with one image of split-MNIST dataset. (b) illustrates the inference energy of the network for various numerical formats.

Table 2: The estimated energy dissipation of the spiking network units during continual learning training.

Module	Bit precision	Energy(pJ)
Accumulator	16-bit	0.2176
Multiplier	16-bit	1.471
LIF Neuron	16-bit	5.2192
Error accumulation	16-bit	1.1202
Exponent function	16-bit	1.0668
Posit decoder Es= 0	8-bit → 16-bit	0.8917
Posit decoder Es= 1	8-bit → 16-bit	0.9932
Posit decoder Es= 2	8-bit → 16-bit	0.9532
Posit encoder Es= 0	16-bit → 8-bit	1.216
Posit encoder Es= 1	16-bit → 8-bit	1.140
Posit encoder Es= 2	16-bit → 8-bit	0.9876
Memory		
SRAM read	16-bit	166.73
SRAM write	16-bit	128.48
SRAM read	8-bit	124.39
SRAM write	8-bit	78.17

Figure 2 provides an overview of the estimated energy dissipation during the training and inference of the continual learning model using a single sample of split MNIST.The comparisons in energy estimations were performed on the network, whose parameters were quantized to 16-bit fixed point, 8-bit fixed point, and three topologies of 8-bit posit formats. Here, the energy was estimated on the basis of specific design choices, where computations were executed using a 16-bit precision, while the parameters were stored at their respective quantization precision. Notably, a significant reduction in training energy requirements is evident in the plot when transitioning from the 16-bit fixed point to all 8-bit representations, which can be attributed to the disparity in access energy between the 16-bit memory and 8-bit memory. However, a slight increase of $\approx 1-2\%$ in energy consumption was observed for posit compared to 8-bit fixed point due to the inclusion of posit encoder and decoder modules. Despite this minor increment in energy, the 8-bit posit-quantized models outperformed the 8-bit fixed point by achieving $\approx$ 30% improvement in mean accuracy. This indicated that with careful quantization, the continual learning model shows robustness to quantization while capitalizing on its benefits. A similar pattern of energy savings through parameter quantization to 8-bit posit compared to 16-bit fixed point, can be observed in inference as well.

5 CONCLUSION

In this paper, we propose a low-cost, memory-efficient quantization technique for continuous learning networks that can be deployed on edge devices. The proposed quantized model utilizes 8-bit quantized metaplasticity and synaptic consolidation techniques to mitigate catastrophic forgetting. Beside mitigating the catastrophic forgetting, the 8-bit quantization reduces the total memory requirement by 2× compared to 16-bit fixed point and 4× compared to 32-bit floating point with marginal degradation in the mean accuracy across tasks. In terms of energy efficiency, we found that the quantized posit network reduces energy consumption by $\approx$ 27% compared to the 16-bit fixed point. These enhancements in computation cost, energy efficiency, and compactness can seamlessly support continual learning on the edge.

ACKNOWLEDGMENTS

This effort is partially supported by the NSF NAIAD Award #2332744 and Air Force Research Laboratory under agreement number FA8750-20-2-1003 through BAA FA8750-19-S-7010. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of NSF, Air Force Research Laboratory or the U.S. Government

REFERENCES

[1] Wickliffe C. Abraham and Mark F. Bear. 1996. Metaplasticity: The Plasticity of Synaptic Plasticity. *Trends in Neurosciences* 19, 4 (1996), 126–130. https://doi.org/10.1016/S0166-2236(96)80018-X

[2] Bernard Ans and Stéphane Rousset. 1997. Avoiding catastrophic forgetting by coupling two reverberating neural networks. *Comptes Rendus de l'Académie des Sciences-Series III-Sciences de la Vie* 320, 12 (1997), 989–997.

[3] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. 2019. Continual learning: A comparative study on how to defy forgetting in classification tasks. *arXiv preprint arXiv:1909.08383* 2, 6 (2019).

[4] Robert M French. 1992. Semi-distributed representations and catastrophic forgetting in connectionist networks. *Connection Science* 4, 3-4 (1992), 365–377.

[5] Robert M French. 2019. Dynamically constraining connectionist networks to produce distributed, orthogonal representations to reduce catastrophic interference. In *Proceedings of the sixteenth annual conference of the cognitive science society*. Routledge, 335–340.

[6] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2022. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*. Chapman and Hall/CRC, 291–326.

[7] John L Gustafson and Isaac T Yonemoto. 2017. Beating floating point at its own game: Posit arithmetic. *Supercomputing frontiers and innovations* 4, 2 (2017), 71–86.

[8] Bing Han, Abhronil Sengupta, and Kaushik Roy. 2016. On the energy benefits of spiking deep neural networks: A case study. In *2016 International Joint Conference on Neural Networks (IJCNN)*. 971–976. https://doi.org/10.1109/IJCNN.2016.7727303

[9] Vedant Karia, Fatima Tuz Zohora, Nicholas Soures, and Dhireesha Kudithipudi. 2022. Scolar: A spiking digital accelerator with dual fixed point for continual learning. In *2022 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1372–1376.

[10] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* 114, 13 (2017), 3521–3526.

[11] Dhireesha Kudithipudi, Anurag Daram, Abdullah M Zyarah, Fatima Tuz Zohora, James B Aimone, Angel Yanguas-Gil, Nicholas Soures, Emre Neftci, Matthew Mattina, Vincenzo Lomonaco, et al. 2023. Design principles for lifelong learning AI accelerators. *Nature Electronics* 6, 11 (2023), 807–822.

[12] Hamed F Langroudi, Vedant Karia, Zachariah Carmichael, Abdullah Zyarah, Tej Pandit, John L Gustafson, and Dhireesha Kudithipudi. 2021. ALPS: Adaptive Quantization of Deep Neural Networks With GeneraLized PositS. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3100–3109.

[13] Hamed F Langroudi, Vedant Karia, John L Gustafson, and Dhireesha Kudithipudi. 2020. Adaptive posit: Parameter aware numerical format for deep learning inference on the edge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 726–727.

[14] Hamed F Langroudi, Vedant Karia, Tej Pandit, and Dhireesha Kudithipudi. 2021. TENT: Efficient Quantization of Neural Networks on the tiny Edge with Tapered FixEd PoiNT. *arXiv preprint arXiv:2104.02233* (2021).

[15] Zhizhong Li and Derek Hoiem. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence* 40, 12 (2017), 2935–2947.

[16] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. 1995. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review* 102, 3 (1995), 419.

[17] Emre O Neftci, Charles Augustine, Somnath Paul, and Georgios Detorakis. 2017. Event-driven random back-propagation: Enabling neuromorphic deep learning machines. *Frontiers in neuroscience* 11 (2017), 324.

[18] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. 2019. Continual lifelong learning with neural networks: A review. *Neural Networks* (2019).

[19] Rachmad Vidya Wicaksana Putra and Muhammad Shafique. 2021. Q-spinn: A framework for quantizing spiking neural networks. In *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[20] Edmund T Rolls and Alessandro Treves. 2011. The neuronal encoding of information in the brain. *Progress in neurobiology* 95, 3 (2011), 448–490.

[21] Kaushik Roy, Akhilesh Jaiswal, and Priyadarshini Panda. 2019. Towards spike-based machine intelligence with neuromorphic computing. *Nature* 575, 7784 (2019), 607–617.

[22] J Rueckl. 1993. Jumpnet: A multiple-memory connectionist architecture. In *Proceedings of the 15 th Annual Conference of the Cognitive Science Society*, Vol. 24. 866–871.

[23] Nicholas Soures, Peter Helfer, Anurag Daram, Tej Pandit, and Dhireesha Kudithipudi. July 2021. TACOS: Task Agnostic Continual Learning in Spiking Neural Networks. In *Theory and Foundation of Continual Learning Workshop at ICML2021*.

[24] Gido M van de Ven, Nicholas Soures, and Dhireesha Kudithipudi. 2024. Continual Learning and Catastrophic Forgetting. *arXiv preprint arXiv:2403.05175* (2024).

[25] Gido M. van de Ven and Andreas S. Tolias. 2019. Three Scenarios for Continual Learning. *arXiv:1904.07734 [cs, stat]* (April 2019). arXiv:1904.07734 [cs, stat]

[26] Friedemann Zenke, Ben Poole, and Surya Ganguli. 2017. Continual learning through synaptic intelligence. In *International conference on machine learning*. PMLR, 3987–3995.

[27] Fatima Tuz Zohora, Vedant Karia, Anurag Reddy Daram, Abdullah M Zyarah, and Dhireesha Kudithipudi. 2021. MetaplasticNet: Architecture with Probabilistic Metaplastic Synapses for Continual Learning. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1–5.

[28] Fatima Tuz Zohora, Abdullah M Zyarah, Nicholas Soures, and Dhireesha Kudithipudi. [n. d.]. Metaplasticity in Multistate Memristor Synaptic Networks. ([n. d.]). arXiv:2003.11638 [cs] https://arxiv.org/abs/2003.11638