Biological Dynamics Enabling Training of Binary Recurrent Networks

G. William Chapman, Corinne Teeter, Sapan Agarwal, T. Patrick Xiao, Park Hays, Srideep S. Musuvathy
Sandia National Laboratories
Albuquerque, New Mexico
Email: {gwchapm*, cmteete, sagarwa, txiao, phays, smusuva}@sandia.gov

2024 Neuro Inspired Computational Elements Conference (NICE) | 979-8-3503-9058-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/NICE61972.2024.10549632

***Abstract*—Neuromorphic computing systems have been used for the processing of spatiotemporal video-like data, requiring the use of recurrent networks, while attempting to minimize power consumption by utilizing binary activation functions. However, previous work on binary activation networks has primarily focused on training of feed-forward networks due to difficulties in training recurrent binary networks. Spiking neural networks however have been successfully trained in recurrent networks, despite the fact that they operate with binary communication. Intrigued by this discrepancy, we design a generalized leaky-integrate and fire neuron which can be deconstructed to a binary activation unit, allowing us to investigate the minimal dynamics from a spiking network that are required to allow binary activation networks to be trained. We find that a subthreshold integrative membrane potential is the only requirement to allow an otherwise standard binary activation unit to be trained in a recurrent network. Investigating further the trained networks, we find that these stateful binary networks learn a soft reset mechanism by recurrent weights, allowing them to approximate the explicit reset of spiking networks.**

***Index Terms*—Recurrent Networks, Spiking Neural Networks, Neuromorphic Computing, Video Processing**

I. Introduction

Many machine learning tasks involve stimuli which evolve in both space and time, such as tracking objects in a video or identifying a scene based on the interaction of actors. Such tasks can sometimes be processed in a sequential spatial-then-temporal approach, by extracting large scale spatial feature information and evaluating how those features evolve through time. However, in other cases the temporal information may be of higher importance and finer spatial scale. For example, when identifying an object from a distance, integrating temporal information may allow one to detect changes when the spatial resolution was too low to otherwise identify an object [1]. In such cases recurrent processing must occur early in the hierarchy, in order to avoid smoothing small temporal signals before they can be extracted. In neural systems, recurrent processing is ubiquitous as early in the visual processing as the retina [2], including hierarchically recurrent processing in which predictions are incorporated into lower-level circuits, possibly increasing sensitivity to small signals [3], [4]. Here we investigate a particular case for recurrent spatiotemporal processing at early layers of a machine learning model, taking inspiration from biological systems in order to train on binary activations, similar to spiking neurons.

Hardware Constraints: While several use cases may seek to utilize recurrent neural networks for processing of information near sensors, such hardware is often restricted by size, weight, and power (SWaP) constraints. Binarized activation neural networks (BANNs) can minimize the precision of analog-to-digital converters, in the case of physical accelerators such as memristor crossbars [5], or otherwise minimizing the number of binary operations required for linear arithmetic. However, many use cases have temporal dynamics, which requires the use of recurrent neural networks, which requires the storage of state and output over time steps. Storing these stateful variables and moving them between memory and compute regions, is energetically expensive. We therefore require a network which incorporates stateful recurrence, but which minimizes the amount of state that must be retained. This can be achieved either by decreasing the number of layers with statefulness, or by decreasing the amount of state stored by each layer. Previous research has shown that for scenarios such as object tracking or video classification, the best performing networks require recurrence at the earliest, largest, layers of processing [6]. Therefore, we seek a method for optimizing recurrent neural networks in which the size of state is minimized by utilizing a single bit activation function.

Training Binary Networks: In recent years training binary, or otherwise heavily discretized neural networks, has been largely achieved by the use of surrogate gradient descent, which approximates the discontinuous activation functions as a continuous functions in the backwards pass [7], [8]. Such approaches can be highly effective, especially when the activation surrogate function closely approximates the activation function. Other approaches have attempted to use activations with tuneable sharpness, such that training starts on a fairly smooth activation, which becomes increasingly close to the binary activation over the course of training [9]. However, neither of these approaches have been successful in training recurrent layers, likely due to feedback activations pushing the units away from the regime where the surrogate functions are valid approximations [10]. However, previous work has been successful in training recurrent spiking networks [11], which we utilize in this work to train a binary activation network.

II. Spatiotemporal Networks for Object Tracking

A. Task

We introduce a small object tracking (SOT) task, designed specifically to test the capability of our networks to detect fine-grained spatiotemporal information. We consider a synthetic task in which a remote camera moves slowly over a large field of view, and on which we superimpose a single target object moving according to known dynamics. Background images are 800x800 pixels, taken from the DIOR dataset [12] consisting of approximately 12,000 images of diverse structure, which are converted to grey-scale. To model sensor movement we crop to a 30x30 sample of the background image and slowly move the center of that window by modeling the sensor velocity as a time-varying Ornstein–Uhlenbeck (OU) process, and interpolate with background translation by cubic splines. Sensor noise is modeled by adding independent white noise on at each pixel, with variance proportional to the intensity of that pixel. Finally, object locations are generated as bounded ballistic trajectories that experience elastic collisions with the boundary of the frame. After adding the noise and target values to the background image, the resulting pixel values are clipped between zero and one. The resulting task is a time-varying input of images, and the target output is the sub-pixel location of the object, with performance measured as mean-squared-error (MSE).

For the results presented here, we normalize the background image to a maximum intensity of 0.5, and the target intensity is 0.25 giving a target-to-background ratio of 0.5. The proportional noise was set to a standard deviation of one-quarter of the pixel intensity, further decreasing the signal to noise ratio and introducing clutter in the temporal difference of images. The mean object velocity is equal to one-third of a pixel per frame, resulting in the object not moving between pixels on the majority of frames, and the trial lasted for 100 frames. All results are presented on a training set of 50,000 trials and a separate validation set of equal size.

B. Model Architecture & Training

As the SOT task is spatiotemporal in nature, we utilize a combination of dense, convolutional, recurrent, and recurrent-convolutional [6] connections, with configurations described below. On each trial images from the tracking task are presented one at at time to an early convolutional layer, and the readout layer consists of two units which are interpreted as the position of the moving object within the field of view. All models are fit to minimize the mean-square error (MSE) of the estimated location on each frame, utilizing backpropagation (through time, as appropriate) and the standard ADAM [13] optimizer. For every network configuration we ran 5 separately initialized models and show their average performance over the course of 200 epochs.

All models begin with a two layer convolutional network with 8 and 16 channels, with the exception of the ”shallow” network, which consists of only a single 8 channel connection. In the case of convolutional-recurrent layers (CRNN), these

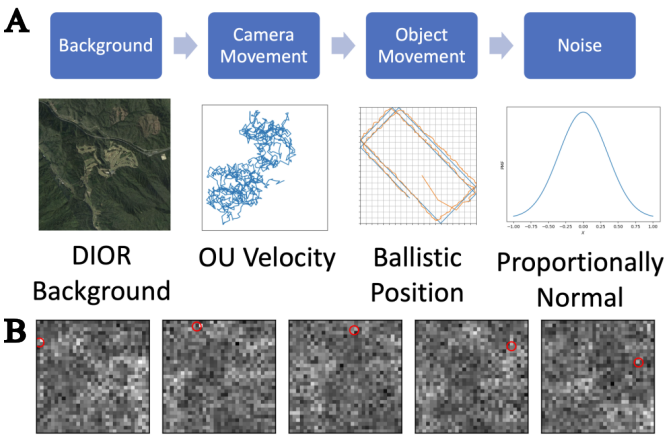

Fig. 1. Process for generation of the small object tracking dataset. **A:** Background images are taken from the DIOR dataset and turned into greyscale, then a small random drift is added to the overall location. The target object location is simulated as ballistic motion, and a small intensity signal is added to the corresponding pixel. Finally, sensor noise is modeled as proportionally normal and added to each pixel in an independent manner on each frame. **B:** Illustrating 5 frames sampled equally from a single trial, with object locations circled in red. On some frames (first and second from the left) the object is clearly visible, while on other frames (3-5 here) it is obscured by the background and noise signal.

convolutional channels had an additional set of weights which map each channel's output as an additional input at the next point in time. Convolutional layers utilized a rectified linear activation, while CRNN layers utilized the hyperbolic tangent. Spatial processing was then followed by two readout layers with 100 and 2 units, implemented either as a rectified linear layer or as a gated-recurrent unit. In order to verify that there was sufficient visual clutter in the stimulus, we also implemented a difference convolutional network (DCNN) which received the temporal change in frames, rather than the raw value.

C. Early Recurrence is Necessary

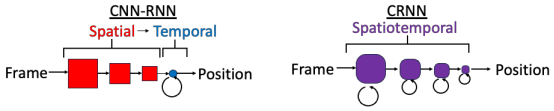

Fig. 2. Left: Sequential CNN-RNN approach, in which spatial information is extracted and reduced, before being processed by a small temporal network. Right: A Convolutional recurrent neural network (CRNN), in which high fidelity spatial and temporal information is processed at every layer.

Table I and figure 3 summarize the performance of the various network configurations. Overall, we find the purely spatial (CNN, DCNN) networks fail to learn past the first few epochs, while spatial-then-temporal networks (CNN-RNN, Figure 2 Left) overfit to specific video-sequences, but fail to generalize to novel samples. In contrast, all networks with a convolutional recurrent layer (CRNN, Figure 2 Right) at the lowest layers of the network perform accurate sub-pixel tracking on both training and validation sets. The single-layer

CRNN ('CRNN-Shallow') performed only marginally worse than the deeper models, while the model with a CNN layer before the CRNN ('CNN-CRNN') had an error three-times as high as the shallow network. These results show that for the SOT task a small degree of spatiotemporal processing may be necessary, but that it must occur early in the network.

Network	Convolutions	Readout	Validation Loss
CNN	2 Conv2D	Linear	.093
DCNN	2 Conv2D	Linear	.290
CNN-RNN	2 Conv2D	GRU	.091
CRNN	2 CRNN	Linear	.007
CRNN-Shallow	1 CRNN	Linear	.006
CNN-CRNN	Conv2D, CRNN	Linear	.018
CRNN-CNN	CRNN, Conv2D	Linear	**.004**

TABLE I
SOT PERFORMANCE ACROSS ALL NETWORK ARCHITECTURES. ONLY NETWORKS WITH CRNN LAYERS PERFORM SUB-PIXEL LOCALIZATION

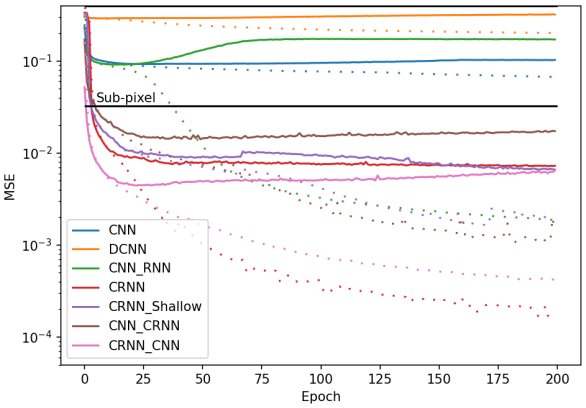

Fig. 3. Continuous-valued network performance over the course of training, on both training data (dashed) and validation data (solid). Spatial-then-temporal models overfit on noise, without generalization to test datasets. Spatiotemporal models generalize to testing data, and perform sub-pixel localization.

III. BINARY RECURRENT NETWORK TRAINING

The previous section demonstrates that early recurrence is necessary for accurate tracking, and that the convolutional-recurrent layer is sufficient to extract the necessary spatiotemporal changes to track the object. While previous results show that binary activation layers can drastically decrease these energy costs [5], previous work has not shown that binary-activation units can be trained in recurrent layers. In contrast, biological neurons communicate utilizing binary spikes in highly recurrent systems, and previous work has shown success in training recurrent spiking networks. We therefore sought to test whether binary activation units or spiking neurons can be trained in the convolutional-recurrent networks from above. Aside from the activation functions and surrogate gradients outlined below, these networks have the same architecture and training methods as for the continuous case.

A. Binary Activation Neural Networks

These models replace the continuous activation function of the ANN models with a binary activation function:

$$\Theta(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ -1 & \text{if } x < 0 \end{cases} \tag{1}$$

Which is notably non-differentiable at the threshold and therefore can not be optimized with standard gradient methods. Instead, as with previous approaches during the backward phase the surrogate gradient function is defined as the straight-through estimator (STE) [8]:

$$\hat{\Theta}(x) = \begin{cases} x & \text{if } |x| \leq 1 \\ sign(x) & \text{if } |x| > 1 \end{cases} \tag{2}$$

Which is continuous within [-1, 1], and the bounding term prevents weights from adjusting when the feed-forward variable is far from the activation threshold. Optimization then continued as normal, except that parameters were updated based on the surrogate gradient rather than the true gradient of the forward activation function.

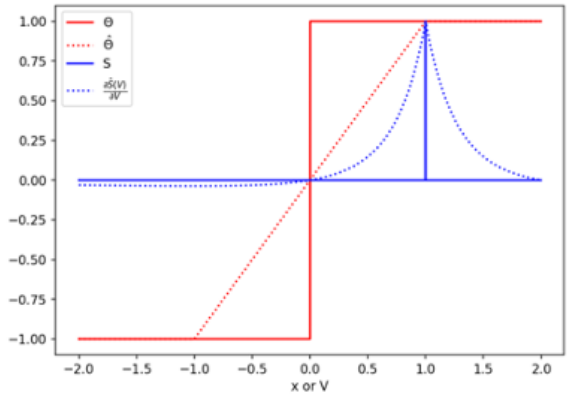

Fig. 4. Activation functions (solid) and surrogate gradients (dashed) for training binary activation networks. (Blue) Spiking neural networks elicit an activation when the state crosses 1, and utilize the derivative of a sigmoid for the surrogate. (Red) Binary networks follow the sign activation and a piece-wise linear straight-through estimator as the surrogate activation.

B. Spiking Networks

Spiking neural networks (SNN) were modeled using Norse [14], a simple simulator built on top of PyTorch utilizing simple forward Euler mechanisms. We utilize leaky-integrate and fire (LIF) neurons, which have previously been shown to successfully train in recurrent networks [15]. Individual units followed the dynamics:

$$\begin{aligned} \tau_v \frac{dv_L(t)}{dt} &= -v_L(t) + \sum_{n \epsilon A} W_{nL} S_n(t) \\ S_L(t) &= (v_L(t) \geq 1) \end{aligned} \tag{3}$$

Where v_L is the sub-threshold voltage, A is the set of all layers projecting to the layer L, W_{nL} is the weights from layer n to L, S is the binary activation, and τ_v is the time constants of the sub-threshold voltage.

With an explicit reset mechanism:

$$v_L(t+1) = \begin{cases} 0 & \text{if } S_L(t) \\ v_L(t) + dv_L(t) & \text{otherwise} \end{cases} \tag{4}$$

The spiking units utilize the "SuperSpike" [16] surrogate gradient, which operates on the subthreshold membrane potential and has the form:

$$\hat{S}(V) = \frac{1}{1 + e^{5*(V-1)}} \tag{5}$$

C. Spiking Enables Binary Recurrent Network Training

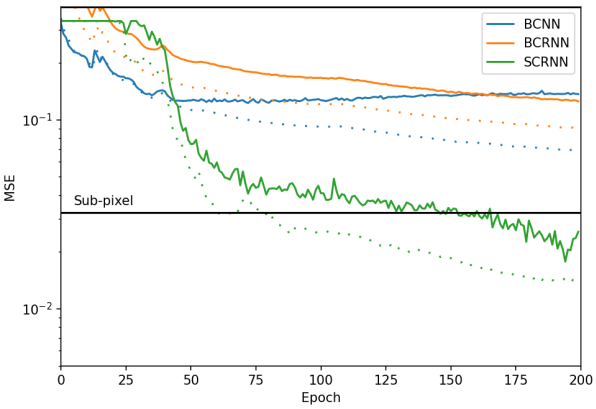

Fig. 5. Performance on binary-activation networks during training. Binary convolutional networks (BCNN) fail to converge, similar to the real-valued CNN. Binary activation recurrent networks (BCRNN) do not train, in contrast to their real-valued counterparts. However, LIF-based spiking convolutional recurrent networks (SCRNN) are able to perform the task at sub-pixel accuracy.

As with the continuous value networks, the binary-activation convolutional network is unable to perform the SOT task (Figure 5, blue). As expected from the lack of previous literature, the recurrent binary activation network is also unable to converge. However, the LIF-based units are able to converge and perform sub-pixel tracking.

IV. Generalized Integrate and Fire Units

There are several differences between the binary activation network and the LIF-based network presented above, which obfuscate the exact mechanism responsible for the LIF-based network training where the binary-activation units do not. Firstly, the LIF-units activate only at a single point, while the binary units have a positive or negative output for all input values. Secondly, the integrator term in the LIF acts as both a low-pass filter smoothing the membrane potential over time. The integrator term also introduces an explicit temporal dynamic to the LIF unit, which the binary activation lacks. Finally, the LIF units contain an explicit reset mechanism.

In order to more fully address which of these aspects is necessary for training of recurrent binary activation networks, we next introduce a slightly expanded generalized leaky-integrate and fire (GLIF) model with the dynamics:

$$\begin{aligned} \tau_v \frac{dv_L(t)}{dt} &= -g_L v_L(t) - g_u u_L(t) + \sum_{n \epsilon A} W_{nL} S_n(t) \\ \tau_u \frac{du_L(t)}{dt} &= -u_L(t) + S_L(t) \\ S_L(t) &= (v_L(t) \geq 1) \end{aligned} \tag{6}$$

With an explicit reset mechanism:

$$v_L(t+1) = \begin{cases} 0 & \text{if } S(t) \text{and Reset} \\ v_L(t) + dv_L(t) & \text{otherwise} \end{cases} \tag{7}$$

This modification introduces an afterhyperpolarization (AHP) term $u(t)$, and tuneable coupling parameters g_L and g_u linking the membrane dynamics to the leak and AHP terms, respectively. The introduction of the AHP term serves two functions. Firstly, it provides a mechanism for a soft-refractory period that the LIF units used in the previous section lack. Secondly, it provides a second route for the previous activity of the unit to propagate through time, such that even if the integrator aspect of the LIF is removed, the AHP from a previous timestep can linger and prevent the unit from becoming active for a short period. Figure 6 illustrates this expanded model, highlighting the multiple feedback mechanisms that may be present.

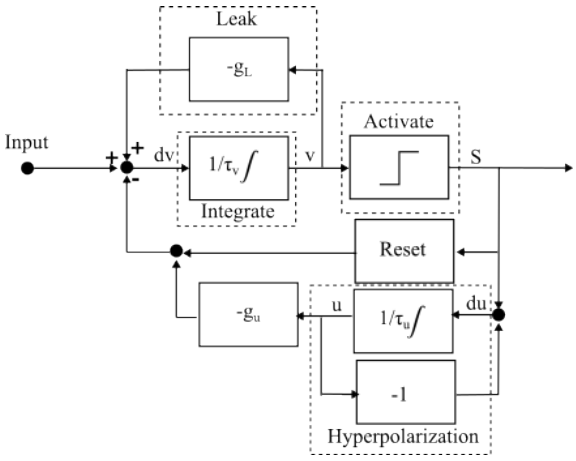

Fig. 6. Block diagram of the components of the spiking network units. By removing various components such as the leak and hyperpolarization term the units can become increasingly simple, and approach the binary-activation units of equation 1, while the full set of blocks implements the generalized model.

Given the GLIF model, we can selectively remove the AHP, integrative, leak, and explicit reset mechanism of the model separately, as highlighted by the variations in table II. A notable difference between the binary and the spiking configurations is that the binary configurations do not have an explicit reset after reaching activation threshold (equation 7).

A. Additional tasks

We next evaluate the variations of the GLIF model when configured into the CRNN network from section II. In addition to this motivating example, which explicitly requires

Model	Abbreviation	g_L	g_u	Reset
Hyperpolarizing LIF	HLIF	1	1	True
Hyperpolarizing Integrate and Fire	HIAF	0	1	True
Leaky Integrate and Fire	LIF	1	0	True
Integrate and Fire	IAF	0	0	True
Binary Activation Leaky Integrate	BLI	1	0	False
Binary Activation Integrate	BI	0	0	False
Hyperpolarizing Binary Activation	HBA	0	1	False
Binary Activation (equation 1)	BA	0	0	False

TABLE II
ALL VARIATIONS OF THE GENERALIZED LINEAR INTEGRATE AND FIRE, ORDERED FROM MOST COMPLETE TO SIMPLEST.

spatiotemporal operations, we test the degree to which the GLIF mechanisms enable or hinder training in other tasks which are more commonly reported. We utilize the MNIST digit dataset [17] to test classification of non-time varying stimuli. This task utilizes 28x28 pixel inputs normalized 0-1, and the output target is a one-hot encoding of labels 0-9, and utilizes the cross-entropy loss function. During training the sample image is presented for 20 sequential time steps and the activity at the readout layer on the last time step is taken to be the output of the network. The MNIST network consisted of two 2-dimensional recurrent-convolutional channels with 8 and 16 channels, each of which is a 3x3 kernel, followed by two readout layers with 100 and 10 units.

To test classification of temporal stimuli, we utilize the free spoken-digit dataset (FSDD) [18], and preprocess the audio files into spectrograms that are 64 time-bins long and 64 frequency bands spanning 0-4Khz. During training the signal for all 64 frequency bands are presented at each time step, and the output layer activity at the final time step is taken as the network output for cross-entropy loss. FSDD networks were the same layout as the MNIST, except that they utilized 1-dimensional convolutions.

B. Binary Recurrent Networks Require State

Unit Type	MNIST	FSDD	SOT (MSE)
Real (CNN-RNN)	98.9	90.4	.091
Real	99.0	98.2	.007
HLIF	98.2	95.6	.015
HIAF	98.4	94.3	.018
LIF	98.5	93.1	.017
IAF	**98.7**	92.7	.016
BLI	**98.7**	**97.2**	**.011**
BI	98.5	93.2	.013
HBA	98.5	31.8	.310
BA	97.2	48.6	.123

TABLE III
PERFORMANCE OF CONVOLUTIONAL RECURRENT ARCHITECTURES WITH VARIOUS UNITS FOR ALL THREE TASKS. BEST BINARY ACTIVATION RESULTS ARE BOLD FOR EACH TASK.

Table III summarizes the performance of the GLIF variations described above, as well as two real-valued baseline models. For all tasks, the binary leaky-integrator performs better than the other models, with the binary-integrator and firing models achieving similar performance. The two models which lack the integrative term (the hyperpolarizing binary, and binary activation) perform well on the MNIST task, but fail on both the spoke-digits and small object-tracking tasks. This suggests that the pre-activation integration of inputs is the primary mechanism responsible for successful backpropagation through time, even when the post-activation hyperpolarization term is available. In models which were unable to properly utilize recurrent connections (HBA, BA), when trained on the MNIST task, those connection weights were minimized (mean-squared amplitude 0.04, compared to 0.78 for BLI), and do not impeded the feedforward pathway from being optimized.

C. Learned Soft-Reset

One notable trend result above is that the reset mechanism does not appear to be necessary for training of these networks, whereas previous work has suggested that reset mechanisms, and particularly the temporal sparsity that they provide, are an essential aspect of spiking neural networks [9]. We therefore calculated the average activity of the models reported in table II, calculated as the proportion of units with an activation on each timestep, and found that the networks had highly similar values (10.7% active for LIF, compared to 11.1% for BLI).

Intrigued by this apparent discrepancy, we investigated the learned recurrent weights in the LIF And BLI networks (see Figure 7). The forward and recurrent weight distributions were not significantly different between the two models, as determined by a Wilcoxon rank-sum test. However, when limited only to the autapses, there is a significant difference between the two models, with LIF autapses being exclusively positive and BLI autapses being exclusively negative. The positive autapses in the LIF model act as a self-excitation, partially restoring the membrane potential following a spike, while the negative autapses of the BLI act as self-inhibition, resulting in a soft-reset once the membrane potential reaches threshold.

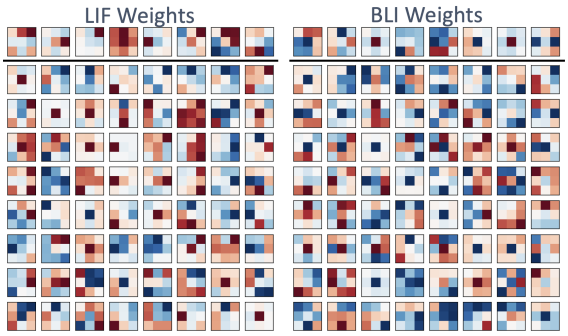

Fig. 7. Learned weights for the feedforward (top) and recurrent (below black line) convolutional kernels of the LIF and BLI networks after training on the SOT task. Both sets of self-recurrent weights (diagonal blocks) contain a consistent self-excitation (LIF) and self-inhibition (BLI) weight.

V. DISCUSSION

Motivated by the capability of nervous systems to detect small objects in low signal-to-noise environments, we investigated the capabilities of recurrent convolutional networks.

By training on the small object tracking task, we find that spatiotemporal processing must occur at the earliest layers of a network in order to accurately detect small changes, consistent with the high degree of recurrence in biological systems. Then, by generalizing the leaky-integrate and fire neuron we are able to show that various forms of binary-activation units can be trained in recurrent networks. The most essential component appears to be the presence of a sub-threshold integrator, while other mechanisms such as explicit post-spike resets and leakage are not necessary.

Model Complexity: We note that in the current work we utilize one of the simplest models of spiking neurons, whereas additional mechanisms are required to fit phenomenological findings from experimental data [19]. While the binary-integrator simplification was sufficient to enable training in this supervised regression task, it is possible that additional mechanisms such as bursting, subthreshold oscillations, and nonlinear dendritic dynamics are necessary for a host of other phenomenon in neural systems, such as attention [20], working memory [21] and continual or unsupervised learning [22], [23]. The current results should be interpreted only as the minimum complexity of units required for supervised surrogate training, and not a dismissal of other neural properties which may have other use-cases.

Hardware Co-Design: A notable aspect of the current findings is that the subthreshold membrane potential appears to be the critical component for enabling binary activation recurrent networks. This has implications for co-design of neuromorphic hardware, which often focuses on feedforward networks without state [24], or implement recurrence in a digital component [25]. However, for hardware which may need to process spatiotemporal data, it is worth considering the hardware constraint tradeoffs with the need to implement recurrent layers. As mentioned above, the current results demonstrate only the base-minimum for spatiotemporal data, whereas other tasks may require additional dynamics. For example, due to device variability it may be desirable to train on-device instead of with surrogate gradient descent [26], and most biological learning rules require additional dynamics such as bursting [22] or long-term activity traces [27].

ACKNOWLEDGMENT

This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. The authors own all right, title and interest in and to the article and is solely responsible for its contents. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this article or allow others to do so, for United States Government purposes. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan https://www.energy.gov/downloads/doe-public-access-plan. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

REFERENCES

[1] T. J. Ma and R. J. Anderson, "Remote sensing low signal-to-noise-ratio target detection enhancement," *Sensors*, vol. 23, no. 6, p. 3314, 2023.

[2] N. Maheswaranathan, L. T. McIntosh, H. Tanaka, S. Grant, D. B. Kastner, J. B. Melander, A. Nayebi, L. E. Brezovec, J. H. Wang, S. Ganguli *et al.*, "Interpreting the retinal neural code for natural scenes: From computations to neurons," *Neuron*, 2023.

[3] R. P. N. Rao and D. H. Ballard, "Predictive coding in the visual cortex: A functional interpretation of some extra-classical receptive-field effects," *Nature Neuroscience*, vol. 2, no. 1, pp. 79–87, Jan. 1999.

[4] L. Itti and P. Baldi, "Bayesian surprise attracts human attention," *Vision Research*, vol. 49, no. 10, pp. 1295–1306, Jun. 2009.

[5] TP. Xiao, WS. Wahby, CH. Bennett, P. Hays, V. Agrawal, MJ. Marinella, and S. Agarwal, "Enabling high-speed, high-resolution space-based focal plane arrays with analog in-memory computing," in *2023 IEEE Symposium on VLSI Technology and Circuits (VLSI Technology and Circuits)*. IEEE, 2023, pp. 1–2.

[6] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving deeper into convolutional networks for learning video representations," *arXiv preprint arXiv:1511.06432*, 2015.

[7] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," *Advances in neural information processing systems*, vol. 29, 2016.

[8] ——, "Quantized neural networks: Training neural networks with low precision weights and activations," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 6869–6898, 2017.

[9] W. M. Severa, C. M. Vineyard, R. Dellana, S. J. Verzi, and J. B. Aimone, "Training deep neural networks for binary communication with the Whetstone method," *Nature Machine Intelligence*, 2019.

[10] S. Ma, D. Brooks, and G.-Y. Wei, "A binary-activation, multi-level weight RNN and training algorithm for ADC-/DAC-free and noise-resilient processing-in-memory inference with eNVM," *IEEE Transactions on Emerging Topics in Computing*, 2023.

[11] J. K. Eshraghian, M. Ward, E. O. Neftci, X. Wang, G. Lenz, G. Dwivedi, M. Bennamoun, D. S. Jeong, and W. D. Lu, "Training spiking neural networks using lessons from deep learning," *Proceedings of the IEEE*, 2023.

[12] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, "Object detection in optical remote sensing images: A survey and a new benchmark," *ISPRS journal of photogrammetry and remote sensing*, vol. 159, pp. 296–307, 2020.

[13] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Jan. 2017.

[14] C. Pehle and J. E. Pedersen, "Norse - A deep learning library for spiking neural networks," Zenodo, Jan. 2021.

[15] B. Cramer, Y. Stradmann, J. Schemmel, and F. Zenke, "The heidelberg spiking data sets for the systematic evaluation of spiking neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 7, pp. 2744–2757, 2020.

[16] F. Zenke and S. Ganguli, "SuperSpike: Supervised Learning in Multilayer Spiking Neural Networks," *Neural Computation*, vol. 30, no. 6, pp. 1514–1541, Jun. 2018.

[17] Y. LeCun, C. Cortes, C. Burges *et al.*, "MNIST handwritten digit database," 2010.

[18] Z. Jackson, C. Souza, J. Flaks, Y. Pan, H. Nicolas, and A. Thite, "Spoken Digit Dataset," 2018.

[19] C. Teeter, R. Iyer, V. Menon, N. Gouwens, D. Feng, J. Berg, A. Szafer, N. Cain, H. Zeng, M. Hawrylycz, C. Koch, and S. Mihalas, "Generalized leaky integrate-and-fire models classify multiple neuron types," *Nature Communications*, vol. 9, no. 1, p. 709, Dec. 2018.

[20] A. Treisman, "Feature binding, attention and object perception," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 353, no. 1373, pp. 1295–1306, Aug. 1998.
[21] M. E. Hasselmo and C. E. Stern, "Mechanisms underlying working memory for novel information," *Trends in Cognitive Sciences*, vol. 10, no. 11, pp. 487–493, Nov. 2006.
[22] A. Payeur, J. Guerguiev, F. Zenke, B. A. Richards, and R. Naud, "Burst-dependent synaptic plasticity can coordinate learning in hierarchical circuits," *Nature Neuroscience*, vol. 24, no. 7, pp. 1010–1019, Jul. 2021.
[23] G. W. Chapman and M. E. Hasselmo, "Predictive learning by a burst-dependent learning rule," *Neurobiology of Learning and Memory*, p. 107826, 2023.
[24] Q. Xia and J. J. Yang, "Memristive crossbar arrays for brain-inspired computing," *Nature Materials*, vol. 18, no. 4, pp. 309–323, Apr. 2019.
[25] C. Li, Z. Wang, M. Rao, D. Belkin, W. Song, H. Jiang, P. Yan, Y. Li, P. Lin, M. Hu *et al.*, "Long short-term memory networks in memristor crossbar arrays," *Nature Machine Intelligence*, vol. 1, no. 1, pp. 49–57, 2019.
[26] Z. Wang, C. Li, P. Lin, M. Rao, Y. Nie, W. Song, Q. Qiu, Y. Li, P. Yan, J. P. Strachan *et al.*, "In situ training of feed-forward and recurrent convolutional memristor networks," *Nature Machine Intelligence*, vol. 1, no. 9, pp. 434–442, 2019.
[27] C. Clopath and W. Gerstner, "Voltage and spike timing interact in STDP - a unified model," *Frontiers in Synaptic Neuroscience*, vol. 2, no. JUL, pp. 1–11, 2010.