

## RESEARCH ARTICLE

# Investigating Biologically Plausible Neural Networks for Reservoir Computing Solutions

CATHERINE MIA SCHÖFMANN<sup>1</sup>, MARIA FASLI<sup>1</sup>, AND  
MICHAEL TAYNNAN BARROS<sup>1</sup>, (Senior Member, IEEE)

School of Computer Science and Electronic Engineering, University of Essex, CO4 3SQ Colchester, U.K.

Corresponding author: Michael Taynnan Barros (m.barros@essex.ac.uk)

This work was supported by the Economic and Social Research Council (ESRC) funded Business and Local Government Data Research Centre at the University of Essex under Grant ES/S007156/1.

**ABSTRACT** While deep learning and backpropagation continue to dominate the field of machine learning in terms of benchmarks and versatility, recent neuroscientific advances shed light on more biologically plausible approaches. Spiking neural networks (SNNs), modelled after action potential dynamics, offer inherent time sensitivity and more efficiency in terms of performance to complexity. While investigating paradigms to support such alternatives, we attempt to answer whether reservoir computing can benefit from a spiking network based implementation with elements of biologically realistic models. This is done by varying both hyper-parameters and reservoir generation approaches and comparing implementations to spot potential improvements. We demonstrate how customized training of SNNs can result in competitive performance levels at lower operational complexity and be readily applied to other paradigms, such as the development of reservoir dynamics.

**INDEX TERMS** Reservoir computing, spiking neural networks, bio-plausible algorithms, leaky integrate-and-fire, LIF.

## I. INTRODUCTION

Among the various forms of computing architecture that support artificial intelligence solutions, reservoir computing (RC) has been demonstrated to offer simple, effective, and flexible models with lower training loads [1], [2]. RC functions as a method of non-linear transformation through a black box reservoir, enabling high performance with a shallow architecture. Its less resource intensive training method for recurrent neural networks is based on output layer configuration as well as non-stationary behaviour [3]. Moreover, it provides suitable intelligent models for energy-deficient devices with minimal resource management, which is exciting for areas of miniaturised and battery-free devices as well [4]. More recently, reservoir computing has enable biological computing system to upscale in terms of computational power [5].

RC models are based on conventional artificial neuron models, such as integrate-and-fire (IF), which are known to

be oversimplified from a biological perspective. Considering that the efficiency of reservoir computing is attributed to its non-stationary behaviour, maintaining simple models for its artificial neurons is a constraint on the system's true efficiency. Moreover, despite being outperformed in mainstream deep learning by models such as Long Short-Term Memory (LSTM), its minimal complexity promises compatibility with popular models of computational neuroscience in the pursuit of bio-realistic learning. We consider bio-realistic learning to be the mechanism for adequately characterising learning losses from natural biological phenomena. Either for the dynamical behaviour of neurons or networks, plausible biological phenomena can be investigated to improve reservoir computing models and pave the way for bridging the gap between neuroscience discovery and artificial intelligence. RC can be the interface that utilises new knowledge of learning and its physiological details and efficiently verifies with RC models if these mechanisms are interesting or not to the artificial intelligence community.

Conventional artificial neural networks (ANNs) derive from Hebbian learning and the backpropagation

The associate editor coordinating the review of this manuscript and approving it for publication was Zhengqing Yun<sup>1</sup>.

algorithm [6], [7]. While Hebbian learning still is the most plausible mechanism to explain neuronal plasticity as the fundamental concept underlying learning (“Cells that fire together wire together”), it is very limited in biophysical terms [8]. Since basic back-propagation inherently violates factors that would make it a viable candidate for bio-realistic learning [9], the spotlight falls on alternatives and proposals for models that use known physiological principles of our Brains. Spiking neural networks [10] for instance, build on the concept of action potentials [11] for information propagation, only registering a signal once a certain threshold, the membrane potential, is crossed by a function modelling its state. Each neuron uses only its own internal states and information passed from its presynaptic neighbour, which is more consistent with biological networks. It also incorporates inherent time-sensitivity, producing a ‘spike train’ output that can be converted to a continuous or categorical result as desired. SNNs started as a more biologically plausible alternative to compressing the number of neurons [12] required in fully-connected models, such as the Multi-Layer-Perceptron (MLP) [13]. They have since achieved best-in-class benchmarks when run on neuromorphic hardware [4] due to their energy efficiency, proving that striving for biological feasibility is a rewarding endeavour. One significant flaw of spiking nets is the current lack of optimized training possibilities. Applying backpropagation would require extensive changes as spike resets are non-differentiable. Successful applications of SNN backpropagation, for instance via surrogate gradient, have been accomplished [14], but their optimised compressing is none the better in terms of biological plausibility.

There is a lack of studies that address the problem of the benefits of biological phenomenon models for reservoir computing, however recognised their value may be. This is important to help us understand what types of RC models are appropriate for certain AI-driven problems and how to optimise them as well. We need to solve the challenge of comparing biologically-plausible and biologically-inspired methods that allow us to characterise the relationship between model biological complexity and its average performance. Biologically plausible models can be defined by neurological phenomena describing neuronal activity linking biochemical molecular processes in the neuron membrane [15]. In this way, instead of oversimplifying binary firing behaviour, models can describe action potentials through a series of phases based on how molecules transpose neuronal membranes [16]. The true computing power of neurons lies within their biochemical molecular processes, and we should seek to investigate how this true power can be translated to their digital artificial intelligence counterpart [17], [18].

In this paper, we attempt to characterise the benefits of biologically plausible RC models by developing a comparison method between them and artificial neuronal models. Both conversion and custom learning approaches were used to train spike models, revealing a tunable trade-off between performance and complexity reduction.

Said implementations are then utilised for SNN based training and nodes in RC reservoir generation. We explore the impact the choice of neuron implementation has and whether attempts to introduce biologically accurate topology, as shown in [3], are worthwhile. Our models incorporate the concept of neuronal types, including both excitatory and inhibitory neuron function. Ultimately, this work attempts to establish alternative, biologically plausible paradigms as a powerful substitute for existing computing frameworks.

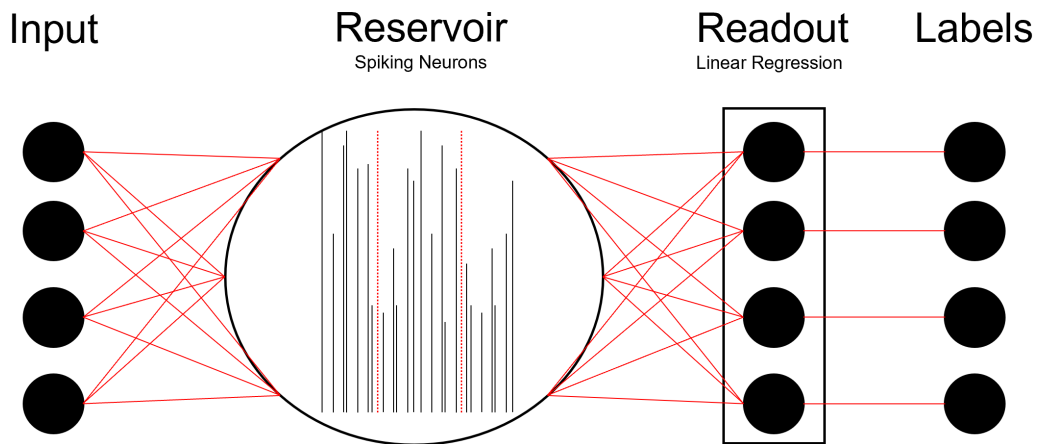
Our contributions are as follows:

- **Evaluation of reservoir computing with spiking nets for reservoir generation** Using spiking representations of poisson transformed image data for non-linear transformation followed by a linear readout, RC is shown to achieve consistent 80 + % accuracy on a 1-pass evaluation of the MNIST hand drawn digit set. This is remarkable for a relatively shallow setup, primarily in regard to its balanced trade-off between performance and model complexity.
- **Investigation of possible enhancements to RC when using biologically sourced data for network dynamics and parameters** It is reasoned that contemporary attempts to connect reservoir dynamics according to topographically accurate connectome models and empirical biological values are unlikely to yield a further increase in performance. However, certain characteristics, such as voltage decay over time and a refractory period, support convergence to the proper output neurons matching MNIST classes in identification.
- **Comparison of modern SNN frameworks** Various spiking net implementations are used to train networks in various ways, comparing not only their benchmarks but also their unique strengths in regards to time sensitivity and complexity reduction. This includes the conversion of a regular convolutional network trained on MNIST to spiking activations with only a 3% drop in accuracy and a reduction in logged operations greater than factor 3.

The primary sections this paper is comprised of are: Literature Review in Section II, Methodology in Section III, Results in Section IV, Discussion in Section V and Conclusion in Section VI.

## II. LITERATURE REVIEW

At the root of the conventional machine learning model lies the universal approximation theorem [19], stating that even a shallow (one layer) feed-forward network can approximate any function given an unbounded number of hidden units. Real world implementations, of course, face conditions and restrictions on their performance. Nonetheless, the advent of deep learning has made scaling to a large number of neurons distributed over layers more feasible and continues to act as the model of choice for competitive performance. A large number of non-recurrent deep networks rely on back-propagation [7], which aims to minimise a loss function for



**FIGURE 1.** Setup for classification tasks using spiking neurons for reservoir generation.

parameters using gradient descent on differentiable functions in order to converge on ideal weights. Back-propagation, however, makes use of weights aggregated and passed back over multiple previous layers, making it incompatible with current assumptions of biological learning, as the latter is said to rely primarily on local neural plasticity.

State-of-the-art models on complex or large-scale datasets often require dedicated hardware not available to the average end-user in order to train in a reasonable time. GPT-3, an autoregressive language model developed by OpenAI, boasts 175 billion parameters [20], taking up hundreds of Gigabytes in storage. For their impressive performance benchmarks, models of this extent can not be considered for use in neuromorphic hardware or any system with severe hardware constraints, such as wearable devices.

Advances towards more plausible learning algorithms are therefore motivated in two ways. In computer science, overcoming these obstacles is critical to increased availability and a reduction of training expenses. Simultaneously, there is a vast research interest in creating biologically sensible machine learning models, promising both insights of neuroscientific value and optimisation through cell properties.

In neuroscience, the most advanced biologically plausible models of neurons can be summarised by the large cross-national efforts, including, most notably, the Blue Brain Project [21]. The project simulated the behaviour of individual neurons and the interactions between them in order to better understand how the brain processes information and generates behaviour within a fully validated framework. To achieve this, they used a combination of experimental data and computational techniques to create a detailed 3D model of the neocortical tissue. The model includes information about the types and properties of individual neurons, as well as the connections between them, which are based on data obtained from electron microscopy and other imaging techniques. Like the idea of this paper, it also takes into

account the electrical and chemical signals that are generated and transmitted between them. This allows researchers to study how the neocortex processes information and generates behaviour under different conditions and in response to different stimuli.

Since the level of biological plausibility can change between different models, from the most detailed in all the spatial-temporal details to the least detailed [22], we decided that for the first step towards a valid comparison, we should most likely decide which models to use based on a specific level of richness. Therefore, in this paper, we investigate models of neural networks that provide richer details of their ionic channel processes and how they describe membrane voltage activity. The model we use as well as the comparison method are detailed in the next sections.

Other ways of developing new artificially intelligent models include the usage of other types of brain cells, including astrocytes [23], [24], and details about neurotransmitter diffusion [25]. In [23], the authors investigated how the synaptic processes of neurons and astrocytes can inspire new Hebbian training rules and demonstrate benefits in multimodal data classification. On the other hand, diffusion AI uses generative models to generate data similar to the data on which they are trained based on thermodynamical physical phenomena [26]. One of the methods destroys training data through the successive addition of Gaussian noise and then learns to recover the data by reversing this process. We see that investigating biologically plausible models is a new class of machine learning that incorporates the methods mentioned above to create a new set of models that particularly utilise biological phenomena for improved AI. We recognise that thermodynamics are a part of plausible biological models; however, they are different from what has already been proposed for diffusion AI. And hence, in this paper, we hope to further highlight the benefits of biologically plausible models.

### III. BIOLOGICAL PLAUSIBLE MODEL WITH SPIKING NEURAL NETWORKS

The most prominent family of biologically realistic models are spiking neural networks (SNN). The idea of spiking activations is reminiscent of cellular mechanisms of brain function, specifically the concept of action potentials for inter-neuronal communication [11]. This process is often modelled using the Hodgkin-Huxley (HH) model, given through Eq. 1, showing the membrane potential  $C_m$  governed by variables for the sodium, potassium, and leak channel conductances, respectively. In the context of machine learning, since a spike is only emitted when the membrane threshold is reached, SNNs are event-based as opposed to the static count of activations in a conventional ANN forward-pass. As spike formation happens with respect to time, SNNs are inherently time-sensitive, taking as input and outputting a series of activations known as spike trains. They can then be converted to a real-valued or classification result by examining their relation to the input at the given time period [27].

$$C_m \frac{\partial V}{\partial t} = -g_{Na}m^3h(V - E_{Na}) - g_Kn^4(V - E_K) - g_L(V - E_L) \quad (1)$$

Neuroscience differentiates the types of neurons between excitatory and inhibitory, which dictates their effect on the firing of an action potential. While excitatory neurons contribute to the opening of sodium channels and thus the firing of a spike, their inhibitory counterparts promote re-polarisation. In respect to machine learning, however, a 1:1 implementation of the empirical biological model proves difficult. Most learning algorithms rely on the easy differentiability of the underlying function to minimise computational cost and retain the interpretability of internal dynamics. This, together with the sheer amount of parameters involved in modelling HH dynamics in each artificial neuron, makes it a possible but currently uncompetitive choice for SNN implementations in regard to train time required.

Recent research has popularised several alternatives to modulate the trade-off between biological accuracy and efficient training. We investigate two of these methods with reservoir computing solutions, the integrate-and-fire (IF) and the Leaky Integrate-and-fire (LIF). Almost all alternate models utilise variations of the integrate-and-fire (IF) model [28], in which specific spike forms are disregarded. This allows them to be encoded as a binary all-or-nothing event, where information is conveyed by either the presence or absence of a spike under certain hyper-parameters, such as the behaviour for potential decay when no input is given. Furthermore, a number of assumptions and approximations are made in the majority of spiking net models. They tend to disregard the difference in excitation type, either treating all neurons as excitatory or assuming an even ratio. Latter is uncommon in real biological systems, where the amount of excitatory neurons far exceeds inhibitory [29].

For modelling neuron states, Leaky Integrate-and-Fire (LIF) neurons are the most common implementation of the

IF mechanic. Leaky neurons introduce a leak current into the membrane potential equation to steadily decay to resting levels  $v_R$ . This leak term is defined through Ohm's law as the difference in potential between the current voltage  $v(t)$  and the resting level over the membrane resistance  $R_m$ . This approximation is in line with biology, as a membrane is not a perfect insulator due to the ionic exchange of potassium and sodium. These ions also interfere with the internal regulation mechanisms of the cell and metabolic processes, including calcium intracellular signalling. In our model, we prefer that all these biophysical mechanisms are just dealt with with membrane resistance for a reasonable comparison with conventional neuronal networks; however, their characterization may lead to further performance improvements if deeply explored. The basic membrane potential formula is shown in 2.

$$C_m \frac{\partial V}{\partial t} = I(t) - \frac{v(t) - v_R}{R_m}. \quad (2)$$

Most implementations additionally make use of a refractory period between spikes in the low millisecond range, during which no new action potential can be formed. Unless data is directly recorded in a time-distributed manner, input to a SNN is encoded to fulfil this requirement. This pre-processing step is most commonly achieved by variations of Poisson-encoding, a distribution method that describes the probability of an independent event happening within a time frame given the mean rate of occurrences. The formula is shown in eq. 3, where  $\lambda$  is the mean rate.

$$P(x) = \frac{\lambda^x}{x!} e^{-\lambda}. \quad (3)$$

#### A. CONVOLUTIONAL NEURONAL NETWORKS

Convolutional Neural Networks (CNNs) are a type of deep neural network that is commonly used for image recognition and classification tasks. In order to draw more analysis from an analytical perspective, we define CNN formally in its conventional form.

Let  $X$  be the input image of dimension  $H \times W \times C$ , where  $H$  is the height,  $W$  is the width, and  $C$  is the number of channels. Let  $Y$  be the output vector of dimension  $K$ , where  $K$  is the number of classes.

The network consists of several layers, including convolutional layers, pooling layers, and fully connected layers. The convolutional layers perform a convolution operation between the input image and a set of filters, also known as kernels or feature detectors.

The convolutional layer's output is as follows:

$$H(i, j, k) = f(W(k) * X(i, j) + b(k)) \quad (4)$$

where  $H(i, j, k)$  is the output of the  $k$ -th filter at location  $(i, j)$  in the output feature map,  $W(k)$  is the  $k$ -th filter,  $X(i, j)$  is the input patch centred at  $(i, j)$ , and  $b(k)$  is the bias term. The function  $f$  is the activation function, such as the rectified linear unit (ReLU) or sigmoid function.

The pooling layers downsample the feature maps by taking the maximum or average value of each local region. This reduces the spatial dimensionality of the feature maps while preserving the important features.

The pooling layer's output is as follows:

$$M(i, j, k) = g(P(H(i, j, k))) \quad (5)$$

where  $M(i, j, k)$  is the output of the  $k$ -th feature map at location  $(i, j)$ ,  $P$  is the pooling function, such as max pooling or average pooling, and  $g$  is an optional activation function.

The fully connected layers perform a linear transformation, then an activation function, on the flattened output of the previous pooling layer:

$$Z = f(W_f M + b_f) \quad (6)$$

where  $Z$  is the output vector of the fully connected layer,  $W_f$  is the weight matrix,  $b_f$  is the bias vector, and  $f$  is the activation function.

Finally, the output layer uses the softmax function to compute the probabilities of each class:

$$Y = \text{softmax}(Z) \quad (7)$$

where softmax is the normalised exponential function.

During training, the weights and biases of the network are adjusted to minimise the cross-entropy loss between the predicted output and the true labels using backpropagation and stochastic gradient descent. The cross-entropy loss measures the difference between the predicted probability distribution and the true probability distribution of the classes.

## B. RESERVOIR MODELS

Reservoir computing is created by parsing the input through a pre-defined reservoir, usually represented by sparsely and randomly connected neurons, as shown in fig. 1.

Let  $X$  be the input vector of dimension  $n$ , and  $Y$  be the output vector of dimension  $m$ . Let  $H$  be the reservoir of  $N$  randomly connected neurons, and let  $W_{in}$ ,  $W_{out}$ , and  $W$  be the input, output, and recurrent weight matrices, respectively. Let  $f$  be the activation function of the neurons.

The following equation can describe the dynamics of the reservoir:

$$H(t) = f(W_{in}X(t) + WH(t-1)) \quad (8)$$

where  $H(t)$  is the state of the reservoir at time  $t$ , and  $H(t-1)$  is its state at time  $t-1$ . The input matrix  $W_{in}$  maps the input vector  $X(t)$  to the reservoir, and the recurrent matrix  $W$  maps the previous state of the reservoir to the current state.

Providing the network's output is:

$$Y(t) = W_{out}H(t) \quad (9)$$

where  $W_{out}$  is the output weight matrix that maps the state of the reservoir to the output vector  $Y(t)$ .

During training, the output weights  $W_{out}$  are adjusted to minimise the error between the predicted output and the target output using a supervised learning algorithm, such as linear regression or ridge regression.

## IV. MODELS TRAINING METHODS

In the following, we propose a new comparison benchmark for the evaluation of biologically plausible and artificial neuronal approaches integrated in a reservoir computing model. We divided the problem into two approaches that deal with the problem of training biologically plausible models. Since neuronal biophysics adds a new level of complexity, training models also need to take into account rules for how far things can go in biology. We describe the setup for all experiments conducted, with their results discussed in the next section.

### A. DATASETS

Whenever applicable, the datasets chosen were MNIST [30] and CIFAR10 [31]. The former is a collection of 70.000 handwritten digits (0 – 9) and is considered relatively simple to achieve high accuracy. It nonetheless serves as a powerful benchmark for whether a network is successfully converging. CIFAR, on the other hand contains 60.000 samples in 10 classes of more complex animals and objects.

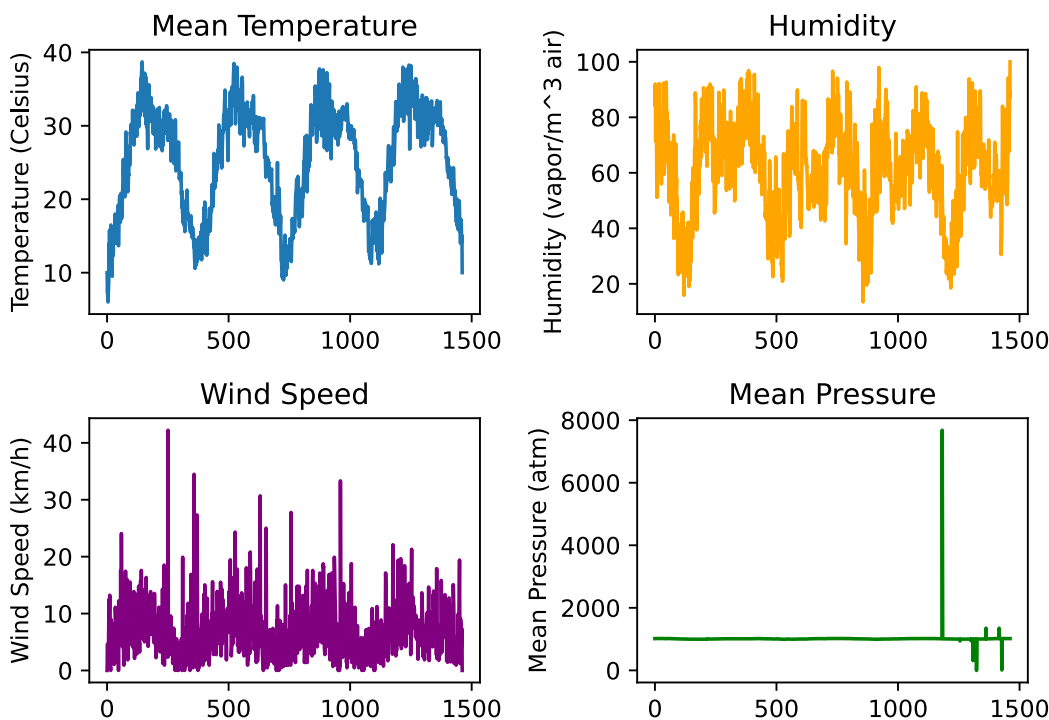
We also use a multi-variate time series problem, depicting weather data in Delhi spanning the years 2013–2017. It was picked due to its periodicity in features and targets to evaluate time-sensitive performance for SNNs. Shown in fig. 2 is a plot of the full data provided in the climate set, where mean temperature is the target variable and predictions for the next day are based on the sliding past two weeks of data each.

### B. CONVERSION TRAINING

Modern approaches have explored the idea of converting pre-trained networks by matching neuron activations to SNN spike firing rates [32]. Reference [32] extends existing work by implementing conversion for layers that are inherently complicated to port, such as pooling and softmax operations, due to their non-trivial nature when working with spikes. The result of this research was a library supporting the creation of SNN conversions for classification tasks built on top of the popular Python ML framework Keras. The compiled model can then be run by a simulator, such as NEURON [33] or the built-in version of INI. Latter was chosen for experiments conducted due to its support of the most common layer types used in a convolutional 2D model. Note that a different approach to pre-processing is taken; instead of poisson-encoding, analogue input is mapped to a constant current. The CNN from Table 1 was first trained on the MNIST dataset for image classification with  $n = 2$  epochs and a batch size of 300. Dense layers use ReLU activations, meaning that the activation value is either 0 or  $x$ . It is then converted using the mean rate matching approach described earlier. For simulation of the converted model via INI, each input is shown for 30ms at a time-step of  $\Delta t = 1$ .

Synaptic operations were logged in order to compare them to the number of floating point operations (FLOPs) in a conventional neural net. A reduction is expected since SNNs only use one addition for weight updates,

### Weather periodicity, entries 2013-2017



**FIGURE 2.** Mean temperature and auxiliary measurements of the Delhi climate dataset plotted over a 4 year period, 2013-2017.

**TABLE 1.** CNN deep-learning architecture for classification of MNIST.

```

Model: "sequential"
-----
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)              (None, 28, 28, 6)          60
max_pooling2d (MaxPooling2D) (None, 14, 14, 6)          0
conv2d_1 (Conv2D)            (None, 14, 14, 3)          165
max_pooling2d_1 (MaxPooling2D) (None, 7, 7, 3)            0
flatten (Flatten)            (None, 147)                  0
dense (Dense)                 (None, 128)                 18944
dense_1 (Dense)               (None, 10)                  1290
-----
Total params: 20,459
Trainable params: 20,459
Non-trainable params: 0
    
```

whereas their regular counterpart requires a multiply-and-accumulate (MAC) operation consisting of addition and multiplication. Non-spiking networks furthermore produce a fixed number of operations per training and simulation step as the feed-forward pass activates each neuron. Spiking nets have the advantage of being event-based; their sparse activations on top of more lightweight updates promise a high factor of compression.

To prove the compression effect, a baseline is taken from the convolutional network (CNN) before it is converted to be compared to synaptic operations after SNN simulation.

The conversion experiment is repeated on the CIFAR10 set to observe potential performance losses on SNN conversion for a complex dataset. For this purpose, an additional combination of conv2d and pooling layers is included in the original architecture.

Even though conversion of pre-trained networks is a fast way to train SNNs that perform similarly well with a smaller net size, it is limited in what it can do and how much it can do. Due to a spiking net’s unique characteristics, certain operations common in conventional ANNs can only be translated suboptimally or not at all. Firing rates are

inherently required to be positive, which causes issues with non-ReLU activation functions for neurons, such as *tanh*, which operates in a  $[-1, 1]$  range. As the conversion is done by rate encoding, SNNs created this way will have their firing rates fixed to approximate their analogue equivalent. They thus lose out on the ability to pick up on spatio-temporal characteristics through unique relationships in spike timing.

### C. CUSTOM TRAINING

A more complete alternative when aiming for temporal abilities is to train SNNs from scratch, using either a modified version of back-propagation or local learning rules. The latter revolves around an adaptation of the Hebbian principle [6], working with spike-time dependent plasticity rules (STDP) [34]. The idea is that the exact timing of spikes in relation to each other determines the reinforcement or deterioration of their synaptic connection. While this approach is far more resource and time consuming for training, it allows for more natural spike dynamics compared to rate-based attempts and can be catered to the individual input data without requiring a certain composition of pre-trained layers.

Over the past years, a multitude of experimental spiking net libraries have surfaced, covering a wide array of use cases and integrating with existing simulators. Among those actively maintained at the time of writing are BindsNET [35] and Norse [36]. They both interface with the PyTorch [37] machine learning framework, a well-established software suite for deep learning in Python. Unlike the previously used conversion library, the focus of these projects is the creation of SNNs from scratch. This allows for both shallow models and combinations with conventional layers in deep learning architecture while still maintaining flexibility in the training approach.

For the first experiment in this category, BindsNET is trained on MNIST for 5 epochs and a batch size of 32 according to [38], which describes a shallow net using 100 excitatory LIF neurons as input and an equal amount of 100 inhibitory counterparts to map output to classes. This is a variation of STDP and is therefore unsupervised. Simulation time is 100ms.

## V. RESULTS

In this section, we present the results for the comparison of spiking networks,

### A. RESERVOIR GENERATION

Using the aforementioned techniques and frameworks for simulation of spiking neurons, various models can be applied for reservoir generation in RC, as described in III-B. In an effort to analyse the merits of applying mappings beyond pure chance to reservoir generation as described in [3], a bare-bones version of an IF neuron is constructed using BindsNet. This is then compared to other results from other neuron cultures, such as regular LIF nodes. For the basic model, a spike is triggered once the cumulative voltage crosses a set threshold, resetting it to a parameter value without

any form of decay or refractory period. The difference in activation and a lack of decay can be seen in the voltage graph in fig. 4. A layer of 100 such neurons is then used as a reservoir following an input layer of 784 nodes, matching the  $28 \times 28$  resolution of the MNIST image set it is evaluated on. The reservoir additionally has one recurrent connection, feeding back processed images. Once generated, the readout layer, consisting of a linear function with sigmoid activation, is trained on the reservoir-parsed representation of 60,000 MNIST samples for a 1 epoch. 10,000 unseen images are reserved for testing, on which all final metrics are based. A batch size of 32 is used. The spike threshold is set to the common value  $-52\text{mV}$ , with a reset value of  $-65\text{mV}$ . As with most SNN models, inputs are first poisson-converted, as shown in fig. 3.

### B. SPIKING NET COMPARISON

For the conversion approach, a validation accuracy of 96.67% was achieved with the traditional convolutional neural network pre-conversion. The transformed SNN subsequently reached 94.67% and 91.67% for ‘reset by subtraction’ and ‘reset to zero’ respectively. Convergence to near conventional ANN levels is achieved after a fraction of the 30ms simulation time, as shown in fig. 5. The total decrease in mean accuracy, therefore, is 2%.

Synaptic operations were logged compared to their regular convolutional counterpart, achieving a reduction of above factor 3 shown in fig. 6. When repeated with a slightly more complex CNN setup as described in Section IV and on the CIFAR10 set, converted accuracy is 59.67% compared to the conventional CNN value of 68.33%. This constitutes a rise in accuracy reduction, totaling 8.66%.

For the custom training experiment, a shallow LIF neuron net was trained using the [38] variation of STDP for 5 epochs, with a batch count of 32. Simulation time was 100ms. Neuron count was set to 100 for both excitatory and inhibitory neurons. It achieves close to 80% accuracy in the first epoch and gradually decays to a mean of 60% at full epoch count.

### 1) TIME-SERIES DATA COMPARISON

For the Delhi time series problem, data was split into a train-test-validation set, each over a sliding window of 14 days to predict the next mean temperature with a batch size of 16. A baseline was then established by predicting the next day’s mean temperature with the value of the previous one. This achieves a Mean Average Error (MAE) of ‘2.08’ degrees on the test set. The convolutional model trained consists of two Conv1D layers followed by a flatten instruction and one fully connected layer.

CNN layers use 8 filters and a kernel size of 3. To analyse whether converted SNNs can pick up on periodicity (Fig. 2) a LSTM model, a conventional recurrent neural net, was initialised with one LSTM layer of 32 units again followed by a dense layer. The loss function used for both was Mean Squared Error (MSE), with MAE logged to plot easily interpretable convergence, shown in fig. 7.

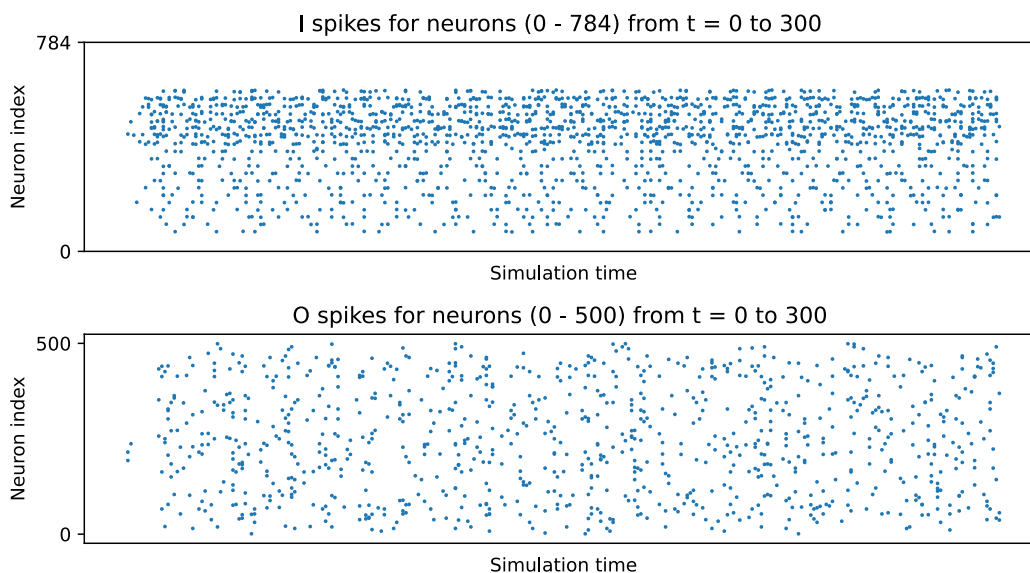


FIGURE 3. Poisson-transformed input spikes compared to reservoir output.

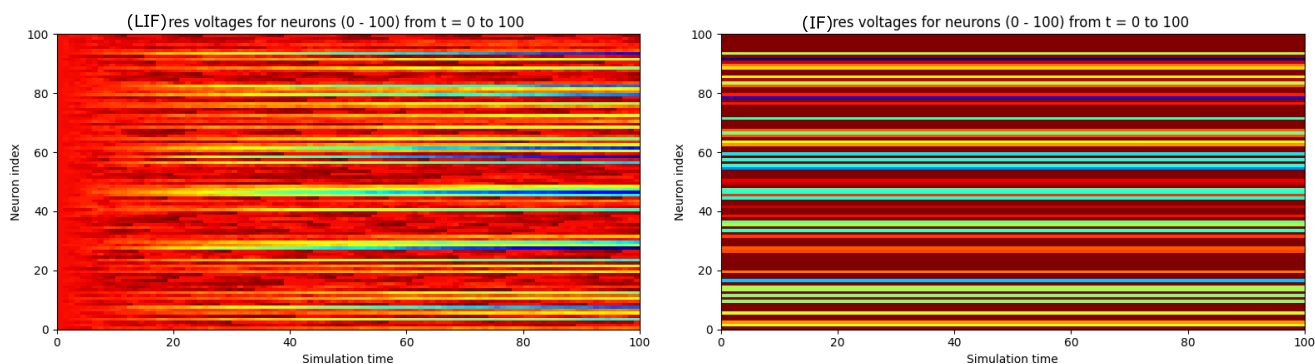


FIGURE 4. “Binarization” of reservoir layer voltage when no decay is given, as seen in the right IF image.

Target epochs were 10 for the CNN and 60 for the LSTM; however, both ran with an early stopping criteria that halts training when a monitored metric stagnates. The recurrent net manages to beat the baseline, halting training at around 40 – 50 epochs with a test-set MAE of 1.37 degrees off the real temperature. The convolutional net only manages a mean error of 3.32 degrees and is therefore below the previous-day dummy approach. Accuracy in modelling the temperature timeline is illustrated in fig. 8.

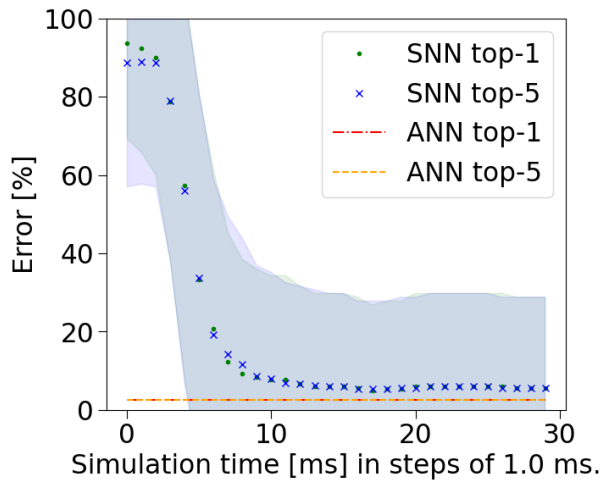
**C. SPIKING RESERVOIR APPLICATION**

Applying the SNN frameworks used in the above comparison, a spiking reservoir of basic IF neurons achieved 68.43% accuracy. When instead using LIF nodes with a refractory period of 5ms and steadily decaying voltage, an accuracy increase of 81.07% is the result. Features of LIF neurons were then selectively added to the simplified IF representation, with the presence of voltage decay alone delivering similar results in the 80% range.

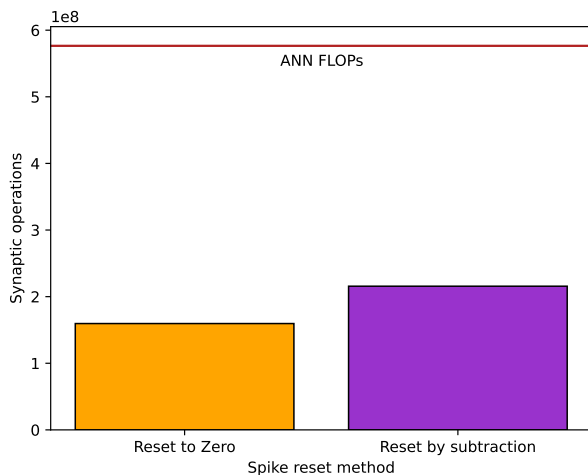
**VI. DISCUSSION**

Over recent decades, biological computing has grown from a novel concept to a research domain with a strong theoretical foundation. Synthetically engineered cells and implants [39] foreshadow alternative treatment and early diagnosis of diseases affecting the brain. At the same time, reconstructions and simulations have increased in amount of features portrayed as well as precision, with the capability of identifying connectivity of even specific neurons at an axonal level [40]. The fields of synthetic biology also promise alleviation for problems silicon-based computation struggles with, such as massively concurrent or time-sensitive operations; both of which are inherent properties of the living cell [41]. Given this rapid development, it is worthwhile revisiting concepts in computer science for possible integration of new paradigms for inter-neuronal communication. Bridging concepts of biologically plausible can also mean that newly synthetically developed biology can also feedback into new forms of artificial intelligence. However, we can already foresee that





**FIGURE 5.** Convergence of error to near conventional ANN levels after a fraction of simulation duration due to a simple source set.



**FIGURE 6.** Synaptic operations for standard (reset by subtraction) and reset by zero approach, the latter incurring a loss in accuracy.

biological computing is a great resource for biocomputing, as evidenced by [5], and needs now more development towards the correct direction. In the following, we showcase results with regard to possible performance improvements of reservoir computing NNs over conventional NNs.

### A. SNN TRAINING COMPARISON

Conversion approaches have been shown to be a convenient method of model compression at a slight loss of performance, which can be alleviated with proper choices in configuration. One such trade-off parameter are reset mechanisms, which determine how membrane potential is reset after a spike event. In graph 6, ‘reset to zero’ shows the simplest case; it does as the name implies and achieves the greatest compression in operations as no additional calculation is required. As MNIST is a relatively simple dataset, the high accuracy is not directly indicative of performance for more complex classification tasks. A higher reduction in performance can

be observed on the CIFAR10 set, increasing 6% compared to MNIST. The settings used take into account the conclusion of ruckauer2017conversion, which recommends analogue activations instead of poisson-converted input to reduce noise. It can be concluded that while passable performance is maintained, scaling the complexity and layers used in the CNN targeted for conversion will lead to increased loss nonetheless. The reduction in operations of roughly factor 3 is owed to SNNs requiring only one addition for state updates whereas regular ANNs employ two through MAC operations, on top of SNNs applying sparse and non-constant activations depending on simulation time-frame. Note that the number of CNN operations reported by the keras profiler and the SNN-conversion logging vary slightly due to different approaches to approximation, but do not alter the ratio shown.

Networks trained from scratch allow for greater control in terms of layer choices and optimisation, as well as inherent time sensitivity. For the 1D time series forecasting on the Delhi climate set, the convolutional network serving as the basis for the converted SNN is limited in layer choices due to the restrictions of the libraries used. This occurs because some operations commonly included in 1D CNNs are either difficult to translate to spiking activations or have not yet been ported from their 2D workaround. The simple approach described nonetheless serves well as a proof of concept for time sensitivity. This is further reinforced when looking at the training graph in fig. 8, showing that even a basic CNN lends itself to overfitting while the recurrent net’s train MAE does not exceed validation. The training MAE of latter interestingly stagnates while validation error further decreases; this could be explained by it picking up on time trends successfully as the validation set split contains the temperature measurements chronologically situated after the train set. Since the spiking net is limited by the convolutional network the conversion is based on, it can not match performance of any conventional time-aware approach.

### B. SNN RESERVOIR GENERATION

The total accuracy for the reservoir model is lower than the previously recorded 90 + % on a converted SNN despite a lower batch size and higher epoch count. However, since the linear readout model is far less complex than any convolutional network, the results are impressive in their own right. It was further shown that certain elements of bio-realistic models, such as passive voltage decay, do have an impact on performance. It essentially removes features from consideration that do not receive any constant input, marking them as irrelevant, as can be seen in the LIF plot of fig. 4. However, the very basic LIF node with which top performance was achieved remains far from more complex biological models. It is unlikely that managing the exact topography of neuron connections would have any positive effect, as Damicelli2021 also found randomness to be key for maintaining performance despite attempts to the contrary. The same goes for the threshold and reset values

MAE Convergence Train-Val compared to Baseline

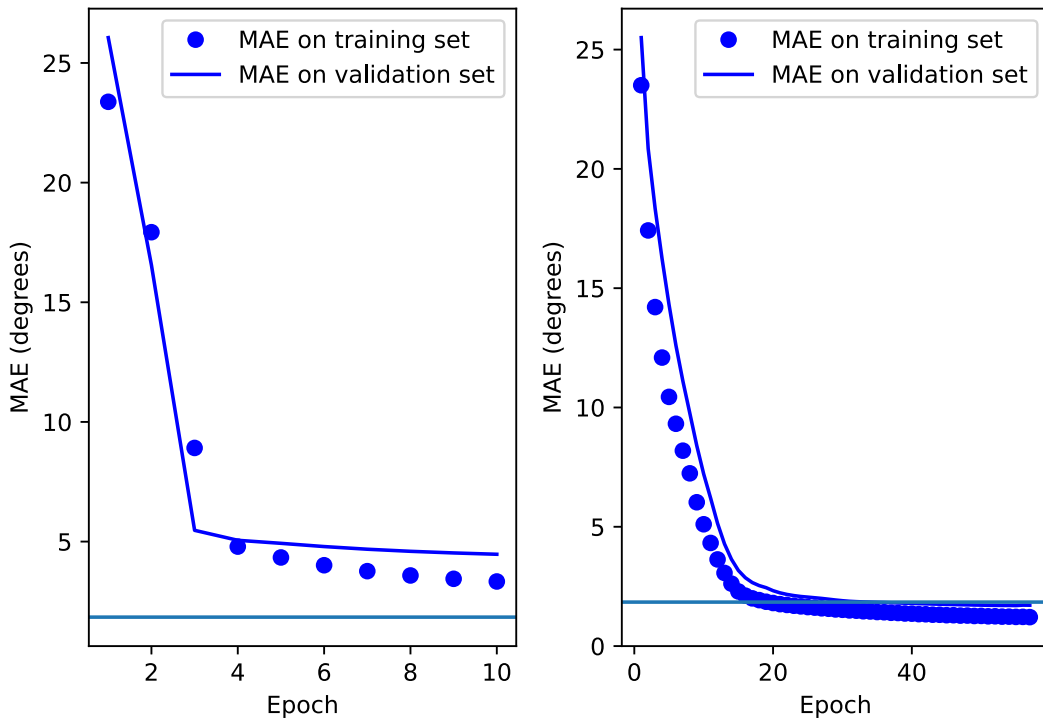


FIGURE 7. Convergence of MAE over training epochs on CNN (left) and LSTM (right), showing the baseline model MAE as a horizontal line.

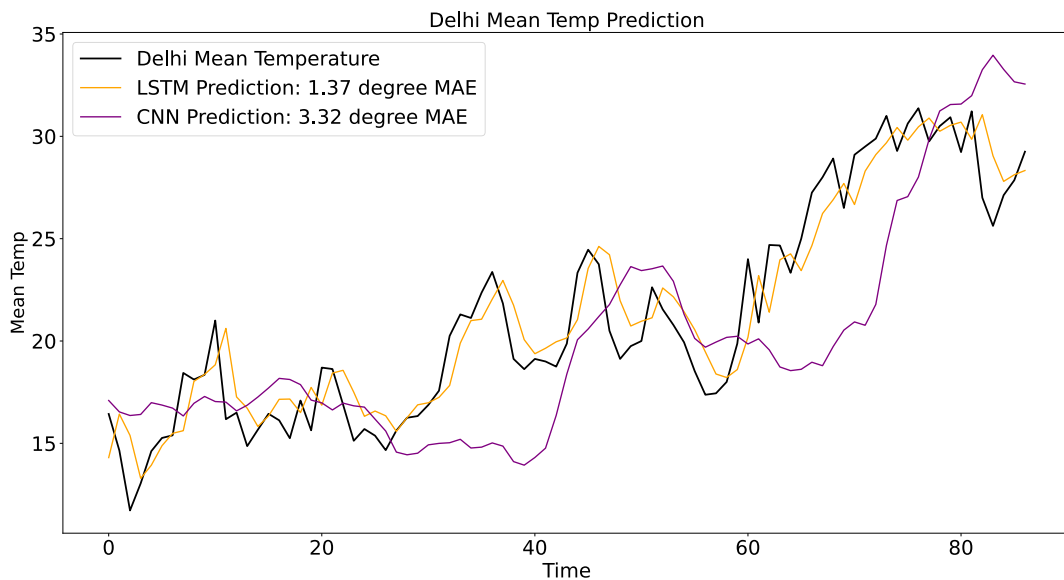


FIGURE 8. Comparison of predicted mean temperatures for the Delhi climate set, showing MAE values in legend.

used;  $-65\text{mV}$  is often cited as an empirical parameter for resting potential in the Hodgkin-Huxley model, yet the exact range matters little for the RC experiment conducted. The same results can be achieved when using values as high as several hundred mV, as long as the cumulative voltage can realistically build up to them within simulation time and

the approximate difference between threshold and reset is maintained.

It stands to reason that reservoir computing for transformation also has potential as part of a bigger deep learning stack, as recent literature confirms [42]. In relation to spiking neural nets, it offers compression due to its shallow layout and

quicker training, but is less versatile than conversion solutions without extensive task-specific setup. Neuron count and other parameters varied in tests, but going above the documented settings tends to cause overfitting and a drop in accuracy.

## VII. CONCLUSION

We showed how bio-realistic reservoir computing alternatives compare to conventional neural networks and conducted experiments to showcase the advantages of individual implementations. Spiking neural networks are great at compressing models because they are based on a few events. This is made even better by the fact that they can be converted from traditional models to save training time. When custom trained they can reach competitive performance levels at lower complexity and easily be utilized for other paradigms such as the creation of reservoir dynamics. Reservoir computing is shown to have synergies with certain characteristics of bio-realistic models but does not stand to profit from over-engineering. It rather remains an underutilised and versatile tool for the evaluation of non-linear problems, no matter the network type underlying reservoir generation.

## APPENDIX

The experiments conducted, as well as the experimental code, may be found at: <https://github.com/otcathatsya/bio-plausible-ml>

## REFERENCES

- [1] H. Jaeger, "The 'echo state' approach to analysing and training recurrent neural networks-with an erratum note," German Nat. Res. Center Inf. Technol. GMD, Bonn, Germany, Tech. Rep., 148, 2001.
- [2] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Comput.*, vol. 14, no. 11, pp. 2531–2560, Nov. 2002.
- [3] F. Damicelli, C. C. Hilgetag, and A. Goulas, "Brain connectivity meets reservoir computing," *PLoS Comput. Biol.*, vol. 18, no. 11, 2022, Art. no. e1010639.
- [4] P. U. Diehl, G. Zarella, A. Cassidy, B. U. Pedroni, and E. Neftci, "Conversion of artificial recurrent neural networks to spiking neural networks for low-power neuromorphic hardware," in *Proc. IEEE Int. Conf. Rebooting Comput. (ICRC)*, Oct. 2016, pp. 1–8.
- [5] H. Cai, Z. Ao, C. Tian, Z. Wu, H. Liu, J. Tchieu, M. Gu, K. Mackie, and F. Guo, "Brain organoid reservoir computing for artificial intelligence," *Nature Electron.*, vol. 6, no. 12, pp. 1032–1039, Dec. 2023.
- [6] D. O. Hebb, *The Organization of Behavior: A Neuropsychological Theory*. New York, NY, USA: Psychology Press, 2005.
- [7] T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink, and D. L. Alkon, "Accelerating the convergence of the back-propagation method," *Biol. Cybern.*, vol. 59, nos. 4–5, pp. 257–263, Sep. 1988.
- [8] R. E. Brown, "Donald O. Hebb and the organization of behavior: 17 years in the writing," *Mol. Brain*, vol. 13, no. 1, pp. 1–28, Dec. 2020.
- [9] S. Bartunov, A. Santoro, B. Richards, L. Marris, G. E. Hinton, and T. Lillicrap, "Assessing the scalability of biologically-motivated deep learning algorithms and architectures," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–11.
- [10] S. Ghosh-Dastidar and H. Adeli, "Spiking neural networks," *Int. J. Neural Syst.*, vol. 19, no. 4, pp. 295–308, Aug. 2009.
- [11] G. B. Ermentrout and D. H. Terman, *The Hodgkin-Huxley Equations*. New York, NY, USA: Springer, 2010, pp. 1–28, doi: [10.1007/978-0-387-87708-2\\_1](https://doi.org/10.1007/978-0-387-87708-2_1).
- [12] W. Maass, "Networks of spiking neurons: The third generation of neural network models," *Neural Netw.*, vol. 10, no. 9, pp. 1659–1671, Dec. 1997. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0893608097000117>
- [13] M. W. Gardner and S. R. Dorling, "Artificial neural networks (the multi-layer perceptron)—A review of applications in the atmospheric sciences," *Atmos. Environ.*, vol. 32, nos. 14–15, pp. 2627–2636, Aug. 1998.
- [14] J. H. Lee, T. Delbruck, and M. Pfeiffer, "Training deep spiking neural networks using backpropagation," *Frontiers Neurosci.*, vol. 10, p. 508, Nov. 2016.
- [15] M. Taynnan Barros, S. Balasubramaniam, and B. Jennings, "Comparative end-to-end analysis of Ca<sup>2+</sup>-signaling-based molecular communication in biological tissues," *IEEE Trans. Commun.*, vol. 63, no. 12, pp. 5128–5142, Dec. 2015.
- [16] G. Basso and M. T. Barros, "Biocomputing model using tripartite synapses provides reliable neuronal logic gating with spike pattern diversity," *IEEE Trans. Nanobiosci.*, vol. 22, no. 2, pp. 401–412, Apr. 2023.
- [17] M. T. Barros, P. Doan, M. Kandhavelu, B. Jennings, and S. Balasubramaniam, "Engineering calcium signaling of astrocytes for neural-molecular computing logic gates," *Sci. Rep.*, vol. 11, no. 1, p. 595, Jan. 2021.
- [18] S. Balasubramaniam, S. Somathilaka, S. Sun, A. Ratwate, and M. Pierobon, "Realizing molecular machine learning through communications for biological AI," *IEEE Nanotechnol. Mag.*, vol. 17, no. 3, pp. 10–20, Jun. 2023.
- [19] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Netw.*, vol. 4, no. 2, pp. 251–257, 1991.
- [20] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, and S. Agarwal, "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1877–1901.
- [21] H. Markram, E. Müller, S. Ramaswamy, M. W. Reimann, M. Abdellah, C. A. Sanchez, A. Ailamaki, L. Alonso-Nanclares, N. Antille, and S. Arsever, "Reconstruction and simulation of neocortical microcircuitry," *Cell*, vol. 163, no. 2, pp. 456–492, 2015.
- [22] M. T. Barros, H. Siljak, P. Mullen, C. Papadias, J. Hyttinen, and N. Marchetti, "Objective supervised machine learning-based classification and inference of biological neuronal networks," *Molecules*, vol. 27, no. 19, p. 6256, Sep. 2022.
- [23] A. B. Porto-Pazos, N. Veigueta, P. Mesejo, M. Navarrete, A. Alvarellos, O. Ibáñez, A. Pazos, and A. Araque, "Artificial astrocytes improve neural network performance," *PLoS ONE*, vol. 6, no. 4, Apr. 2011, Art. no. e19109.
- [24] A. Ahtiainen, B. Genocchi, J. M. A. Tanskanen, M. T. Barros, J. A. K. Hyttinen, and K. Lenk, "Astrocytes exhibit a protective role in neuronal firing patterns under chemically induced seizures in neuron-astrocyte co-cultures," *Int. J. Mol. Sci.*, vol. 22, no. 23, p. 12770, Nov. 2021.
- [25] B. Sun, T. Guo, G. Zhou, S. Ranjan, Y. Jiao, L. Wei, Y. N. Zhou, and Y. A. Wu, "Synaptic devices based neuromorphic computing applications in artificial intelligence," *Mater. Today Phys.*, vol. 18, May 2021, Art. no. 100393.
- [26] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *Proc. ICML*, 2015, pp. 2256–2265.
- [27] M. Pfeiffer and T. Pfeil, "Deep learning with spiking neurons: Opportunities and challenges," *Frontiers Neurosci.*, vol. 12, p. 774, Oct. 2018.
- [28] A. N. Burkitt, "A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input," *Biol. Cybern.*, vol. 95, no. 1, pp. 1–19, Jul. 2006.
- [29] K. T. Sultan and S.-H. Shi, "Generation of diverse cortical inhibitory interneurons," *Wiley Interdiscipl. Rev., Develop. Biol.*, vol. 7, no. 2, p. e306, 2018.
- [30] L. Deng, "The MNIST database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
- [31] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2009.
- [32] B. Rueckauer, I.-A. Lungu, Y. Hu, M. Pfeiffer, and S.-C. Liu, "Conversion of continuous-valued deep networks to efficient event-driven networks for image classification," *Frontiers Neurosci.*, vol. 11, p. 682, Dec. 2017.
- [33] M. L. Hines and N. T. Carnevale, "The NEURON simulation environment," *Neural Comput.*, vol. 9, no. 6, pp. 1179–1209, Aug. 1997, doi: [10.1162/neco.1997.9.6.1179](https://doi.org/10.1162/neco.1997.9.6.1179).
- [34] G.-Q. Bi and M.-M. Poo, "Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type," *J. Neurosci.*, vol. 18, no. 24, pp. 10464–10472, Dec. 1998.

- [35] H. Hazan, D. J. Saunders, H. Khan, D. Patel, D. T. Sanghavi, H. T. Siegelmann, and R. Kozma, “BindsNET: A machine learning-oriented spiking neural networks library in Python,” *Frontiers Neuroinform.*, vol. 12, p. 89, Dec. 2018, doi: 10.3389/fninf.2018.00089.
- [36] C. Pehle and J. E. Pedersen, “Norse—A deep learning library for spiking neural networks,” Jan. 2021. [Online]. Available: <https://norse.ai/docs/> and <https://doi.org/10.5281/zenodo.4422025>
- [37] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, and A. Desmaison, “PyTorch: An imperative style, high-performance deep learning library,” in *Proc. Adv. Neural Inf. Process. Syst.*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, Eds. New York, NY, USA: Curran Associates, 2019, pp. 8024–8035. [Online]. Available: <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>
- [38] P. U. Diehl and M. Cook, “Unsupervised learning of digit recognition using spike-timing-dependent plasticity,” *Frontiers Comput. Neurosci.*, vol. 9, p. 99, Aug. 2015.
- [39] G. L. Adonias, A. Yastrebova, M. T. Barros, Y. Koucheryavy, F. Cleary, and S. Balasubramaniam, “Utilizing neurons for digital logic circuits: A molecular communications analysis,” *IEEE Trans. Nanobiosci.*, vol. 19, no. 2, pp. 224–236, Apr. 2020.
- [40] M. N. Economo, J. Winnubst, E. Bas, T. A. Ferreira, and J. Chandrashekar, “Single-neuron axonal reconstruction: The search for a wiring diagram of the brain,” *J. Comparative Neurol.*, vol. 527, no. 13, pp. 2190–2199, Sep. 2019.
- [41] L. Grozinger, M. Amos, T. E. Gorochowski, P. Carbonell, D. A. Oyarzun, R. Stoof, H. Fellermann, P. Zuliani, H. Tas, and A. Goñi-Moreno, “Pathways to cellular supremacy in biocomputing,” *Nature Commun.*, vol. 10, no. 1, p. 5250, Nov. 2019.
- [42] C. Gallicchio, A. Micheli, and L. Pedrelli, “Deep reservoir computing: A critical experimental analysis,” *Neurocomputing*, vol. 268, pp. 87–99, Dec. 2017.



**CATHERINE MIA SCHÖFMANN** received the M.Sc. degree in computer science from the University of Essex, U.K., in 2022. She is currently pursuing the Ph.D. degree in neuromorphic software ecosystems with the Jülich Research Centre, Germany. Her wider research interests include computational neuroscience, hardware accelerators, and machine learning.



**MARIA FASLI** received the B.Sc. degree in informatics in Greece, and the Ph.D. degree in computer science from the University of Essex, in 2000. Her Ph.D. work was on computational logics for intelligent systems. She is the Director of the Institute for Analytics and Data Science, University of Essex, where she is leading the institute’s development into a pan-university multidisciplinary and interdisciplinary institute covering different aspects of data science, analytics, and big data. She is also the Executive Dean of the Faculty of Science and Health, University of Essex. She has published over 130 articles in the field of AI and data science. She has led, managed, and contributed to a number of projects in these fields. Her research interests include artificial intelligence (AI) techniques and modeling structured/unstructured data. She served as the UNESCO Chair in Analytics and Data Science, whose aim is to work with international collaborators and partners to develop research capacity and skills in data science and analytics for the benefit of society.



**MICHAEL TAYNNAN BARROS** (Senior Member, IEEE) received the Ph.D. degree in computer science from South East Technological University, Ireland, in 2016. He has been an Assistant Professor (a Lecturer) with the School of Computer Science and Electronic Engineering, University of Essex, U.K., since June 2020. He is also the Head of the Unconventional Communications and Computing Laboratory. He was a PI and a Co-I in several grants winning a total of more than 1.5 M funded by the European Commission, BBSRC, Innovate U.K., and others. He has over 80 research peer-reviewed scientific publications in top journals and conferences, such as *Nature Scientific Reports*, *IEEE TRANSACTIONS ON COMMUNICATIONS*, *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, in the areas of molecular and unconventional communications, biomedical engineering, bionano science, and beyond 5G. He is a member of EPSRC and BBSRC Engineering Biology Peer Review Colleges and a fellow of the Higher Education Academy. He received the CONNECT Prof. Tom Brazil Excellence in Research Award, in 2020. He is an Associate Editor of *IEEE TRANSACTIONS ON MOLECULAR, BIOLOGICAL AND MULTI-SCALE COMMUNICATIONS*; and a Review Editor of the *Frontiers in Communications and Networks* in the area of unconventional communications.

• • •