Neurocomputing 464 (2021) 83–94

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Towards understanding the effect of leak in Spiking Neural Networks

Sayeed Shafayet Chowdhury *,[1], Chankyu Lee *,[1], Kaushik Roy

Purdue University, West Lafayette, IN 47907, USA

ARTICLE INFO

Article history:
Received 7 September 2020
Revised 6 July 2021
Accepted 29 July 2021
Available online 8 August 2021
Communicated by Zidong Wang

Keyword:
Leaky-integrate-and-fire (LIF) neuron
Noise robustness
Frequency component
Low-pass filtering
Sparsity

ABSTRACT

Spiking Neural Networks (SNNs) are being explored to emulate the astounding capabilities of human brain that can learn to perform robust and efficient computations with noisy spikes. A variety of spiking neuron models have been proposed to resemble biological neuronal functionalities. The simplest and most commonly used among these SNNs are leaky-integrate-and-fire (LIF), which contain a leak path in their membrane potential and integrate-and-fire (IF), where the leakage path is absent. While the LIF models have been argued as more bio-plausible, a comparative analysis between models with and without leak from a purely computational point of view demands attention, which we try to address in this paper. Our results reveal that LIF model provides improved robustness and better generalization compared to IF. Frequency domain analysis demonstrates that leak aids in eliminating high-frequency components from the input, thus enhancing noise-robustness of SNNs. Additionally, we compare the sparsity of computation between these models. In general, for the same input, the LIF model would be expected to achieve higher sparsity compared to IF due to the layer-wise decay of spikes caused by membrane potential leak with time. However, contrary to this expectation, we observe that leak decreases the sparsity of computation. Therefore, there exists a trade-off between robustness and energy-efficiency in SNNs which can be optimized through suitable choice of amount of leak in the models.

1. Introduction

Over the past few years, the advancements of deep artificial neural networks (ANNs) have led to remarkable success in various cognitive tasks (*e.g.*, vision, language and behavior). In some cases, neural networks have outperformed the conventional algorithms and achieved human-level performance [1,2]. However, recent ANNs are extremely compute-intensive [3] and often suffer from severe accuracy degradation if the testing data is corrupted with noise [4], which was unseen during training. On the other hand, human brain can reliably learn and compute intricate cognitive tasks with only a few watts of power budget. Inspired by this, Spiking Neural Networks (SNNs) have been explored recently towards realizing robust and energy-efficient machine intelligence taking cues from neuroscience experiments [5].

SNNs are categorized as the new generation neural networks [6] based on their neuronal functionalities. A spiking neuron integrates the inputs over time and fires a spike-output whenever the membrane potential exceeds a threshold. A variety of spiking neuron models largely resemble biological neuronal mechanisms, which transmit information through discrete spatio-temporal events (or spikes). These spiking neuron models can be characterized by their internal state called the membrane potential. There exists a range of neuron models which vary in model complexity and degree of capacity to emulate biological neuronal behavior [7–9]. Among them, the Hodgkin–Huxley model [10] lies at the high end of the complexity spectrum, which can capture the interrelation between various biophysical observables. However, for SNN architectures comprising of a large number of neurons, it becomes challenging to find the right set of parameters for such complex models. Additionally, the computational power required to simulate sophisticated neural models with multiple differential equations (*e.g.*, Hodgkin–Huxley [10], Izhikevich [11]) also increase greatly [12]. Because of these reasons, there is a lack of understanding of how each of the factors determining the biological neuronal response can be effectively used in learning. On the other hand, the simplified and computationally compact leaky-integrate-and-fire (LIF) model provides a suitable compromise between complexity and analytical tractability when implemented for large neural networks. In this context, Teeter et al. [12] report various important aspects of using LIF neurons in computational models including satisfactory level of biologically realistic behavior, linearity of the dynamical equations and ease of interpretability. In addition to model complexity, another aspect of choosing LIF

* Corresponding authors.
E-mail addresses: chowdh23@purdue.edu (S.S. Chowdhury), lee2216@purdue.edu (C. Lee).
[1] These authors contributed equally.

https://doi.org/10.1016/j.neucom.2021.07.091
0925-2312/

models is the availability of learning algorithms for large scale SNNs. Several recent works [13–15] have successfully implemented surrogate gradient methods using LIF models for training deep SNN. However, training multi-layer SNNs using the complex neuron models (*e.g.*, Hodgkin–Huxley [10], Izhikevich [11]) remains a challenge till date. It is difficult to investigate the effect of leak if we cannot train models adequately and figure out the role played by the leak parameter. Hence, in this work, we focus on computational models for spiking neurons which either use a Leaky Integrate and Fire (LIF) model with a built-in leaky behavior in the membrane potential, or the simpler Integrate and Fire (IF) with no leak in the membrane potential [16]. Reference [17] proposed a temporal coding scheme based on spike-timing of neurons using IF neurons to avoid the problem of the non-differentiability of spike activation function. Although it achieved competitive performance on MNIST, custom constraints on synaptic weights and a gradient normalization strategy had to be adopted to overcome the problems of dead neuron and gradient explosion, respectively. Meanwhile, IF neuron based time-to-first-spike coding scheme was proposed in [18] to obtain satisfactory results on MNIST. However, these methods focused on shallow networks consisting of linear layers only. Zhang et al. [19] introduced Rectified Linear Postsynaptic Potential function (ReL-PSP) as a new IF neuron model to overcome the challenges of training deep SNNs. The ReL-PSP based model achieved state-of-the-art performance on MNIST using deep convolutional SNNs. Moreover, the researchers in [19] were able to obtain high energy-efficiency due to sparsity of spikes.

The IF and LIF models differ in terms of temporal model dynamics of the membrane potential, for the IF model, this potential maintains a steady state unless new inputs are received, whereas for the LIF model, at each timestep, the potential decays at a certain rate. As a result, some component of input decays out over time, however it is pivotal to investigate the consequences of such membrane potential decay with respect to model performance in classification under noisy conditions and energy-efficient computation. Since biological neuron models have been reported to contain leak (*e.g.*, sodium ion-channel leak [20]), it would be important to quantitatively analyze the advantages and disadvantages of using leaky behavior. Furthermore, the choice of LIF model introduces minimal risk of obscuring important synaptic effects and provides a good basis for comparison as it is widely used as a standard neural model [21].

To that end, we focus on two aspects of the leak effect on LIF SNN models: robustness and spiking sparsity. Ideally, the neural network models are expected to predict reliable outcomes for unseen or even noisy data under sparse spiking events. In addition, compared to ANNs, the main advantage of SNNs is the energy-efficient event-based computing capability, in which the synaptic operations occur only when spike-inputs arrive. To that effect, the computational efficiency of SNNs considerably improves as spike signals become sparser for specialized SNN hardware platforms such as TrueNorth [22] and Loihi [23]. Keeping these two key performance metrics in mind (namely robustness and spike signal sparsity), in this paper, we present a comprehensive and comparative analysis between models with and without leak to delve deeper into the role that leak plays in learning. The main contributions of this work are-

- A theoretical analysis of the first-order phenomenological LIF neuron model is introduced to investigate its low-pass filtering effect. As a step toward this goal, from frequency domain analyses, we show that the presence of leak helps to cut-off some of the input components beyond a certain frequency, thereby aiding the networks to predict more robust outcomes for noisy spike-inputs.
- We examine the effect of leak on computational requirements in multi-layered SNNs. Compared to SNNs with IF model, the ones with LIF model converges with decreased sparsity of spike signals when trained with surrogate-gradient based backpropagation, resulting in reduced computational efficiency.
- We conduct experiments to validate the robustness of multi-layered SNNs with IF and LIF neuron models using popular vision datasets including SVHN and CIFAR-10. Furthermore, we analyze the improved performance of LIF models by investigating the frequency spectrum of spikes and how well the network generalizes to previously unseen data.

2. Spiking Neural Network fundamentals

2.1. Spiking neuron model

The spiking neurons (generally modeled as IF or LIF) are fundamental units in SNNs. The sub-threshold dynamics of an LIF neuron is governed by

$$\tau_m \frac{dU}{dt} = -(U - U_{rest}) + RI, \quad U \leqslant V_{th} \tag{1}$$

where U is the membrane potential, I denotes the input current that represents the weighted summation of spike-inputs, τ_m indicates the time constant for membrane potential decay, R represents membrane leakage path resistance and U_{rest} is resting potential. Fig. 1 depicts the dynamics of LIF neuron and an equivalent circuit model. The input current is accumulated in the membrane potential that decays exponentially over time. The degree of exponential decay is determined by the membrane time constant, $\tau_m = RC$, where R represents resistance of membrane leakage path and C is the membrane capacitance. When the membrane potential exceeds the firing threshold (V_{th}), the neuron is triggered to emit an output-spike and resets the membrane potential to the resting state. The spike-output can be represented as

$$O[t] = \begin{cases} 1, & if\ U[t] > V_{th} \\ 0, & otherwise \end{cases} \tag{2}$$

where $O[t]$ and $U[t]$ denote the spike-output and the membrane potential, respectively, at time instant t. Again, the subthreshold dynamics of an IF neuron can be modeled as

$$C\frac{dU}{dt} = I, \quad U \leqslant V_{th} \tag{3}$$

Note, unlike the LIF case, the IF neuron does not contain any leakage path, so, the potential accumulates with each new incoming spike in a stair-case fashion without decay till it reaches the threshold. Upon reaching V_{th}, the potential is reset and accumulation begins a new cycle.

From a electrical circuit perspective, the neuronal dynamics in Eq. (1) can be represented by an equivalent RC circuit model [24] as illustrated in Fig. 1(c). The parallel RC branch acts as a low-pass filter [25], which has the membrane time constant (τ_m) equal to RC. The analysis of neuronal responses with respect to various frequency components stimulates the following discussion in the next subsection.

2.2. Frequency domain analyses

In this subsection, the response of an LIF neuron model is analyzed in relation to the membrane time constant (τ_m). We investigate the role of leaks in filtering out some of the signal components in the high-frequency range when driven by white Gaussian noise. In order to quantify the low-pass filtering effect, we employ the coherence function, $C(\omega)$ which is a commonly used metric in signal processing [26] to estimate the power transfer from the input

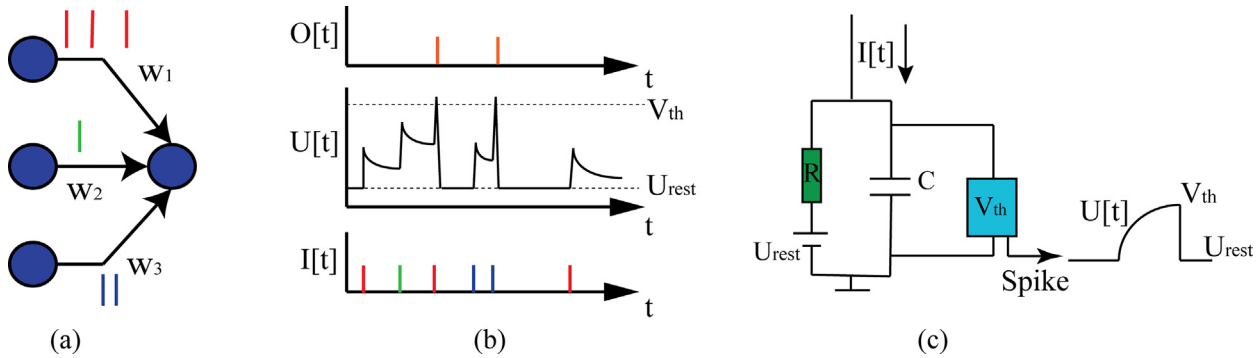


Fig. 1. An LIF neuron, (a) a schematic connection between three pre-neurons to one post-neuron, (b) temporal dynamics of membrane potential in the post-neuron, (c) equivalent circuit model of the LIF neuron.

to the output. When the input to a system is $s(t)$ and the corresponding output is $x(t)$, the coherence between them is defined as

$$C_{x,s}(\omega) = \frac{|S_{x,s}(\omega)|^2}{S_{x,x}(\omega)S_{s,s}(\omega)}, \tag{4}$$

where $S_{x,s}(\omega)$ is the cross-spectrum of output (x) with input (s), $S_{x,x}(\omega)$ and $S_{s,s}(\omega)$ are the autopower spectrum of $x(t)$ and $s(t)$, respectively. To study the response of the neuron model described by Eq. (1), we measure the coherence as a function of frequency. We model the inputs to the neuron as white Gaussian noise current and derive the corresponding coherence between the noise input and the output spike train. The resulting coherence function $C_{x,s}(\omega)$ is given as

$$C_{x,s}(\omega) = \frac{2D_{st}}{D}\frac{r_0\omega^2}{1+\omega^2}\frac{\left|\mathscr{D}_{i\omega-1}\left(\frac{\mu-V_{th}}{\sqrt{D}}\right) - e^{\Delta}\mathscr{D}_{i\omega-1}\left(\frac{\mu-U_{rest}}{\sqrt{D}}\right)\right|^2}{\left|\mathscr{D}_{i\omega}\left(\frac{\mu-V_{th}}{\sqrt{D}}\right)\right|^2 - e^{2\Delta}\left|\mathscr{D}_{i\omega}\left(\frac{\mu-U_{rest}}{\sqrt{D}}\right)\right|^2}, \tag{5}$$

where D_{st} is the intensity of the white noise stimulus, D is total noise intensity (for our case D=D_{st}), r_0 is the output firing rate, $\mathscr{D}(x)$ is a parabolic cylinder function, μ is a parameter denoting DC part of the input (defined in appendix Section A.1) and $\Delta = \frac{U_{rest}^2 - V_{th}^2 + 2\mu(V_{th}-U_{rest})}{4D}$. The detailed derivation of Eq. (5) starting with Eqn. 1 is provided in the appendix Section A.

To analyze the frequency responses of the neuron model, the coherence functions of the IF and the LIF models with high and low leak cases in relation to frequency (ω) are plotted in Fig. 2. This figure shows that the IF model (green) transmits all input components to the outputs across the entire frequency spectrum. On the other hand, for LIF models (red and blue), the coherence function decreases as the frequency increases, thereby cutting-off the high-frequency components propagating to the output. Hence, contrary to the IF model, the LIF model can negate the noise input components beyond a certain frequency limit. Similar low-pass filtering of information for LIF neurons has been reported in [25,27]. The authors in [28] also discussed similar characteristics of LIF neurons from a neuroscience perspective. Drawing inspirations from such phenomenon, our next goal is to explore whether the low-pass filtering effect can enable multi-layered SNNs with LIF neurons to be more robust against noisy inputs. The following subsections focus first on the training methodology adopted in this work, followed by the noisy input generation methods and corresponding experiments.

2.3. Gradient descent learning in SNNs

The gradient-based method, namely backpropagation (BP) learning [29], is a widely employed method for training traditional

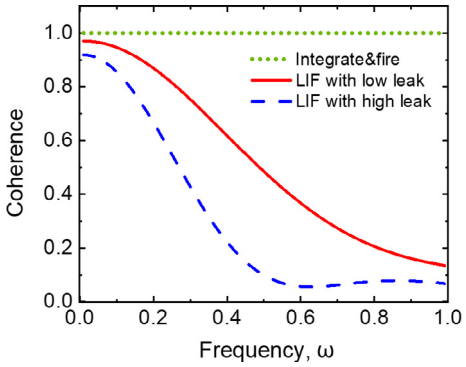


Fig. 2. Illustration of frequency response for IF and LIF neuron models. The horizontal and vertical axes represent the frequency components and coherence function, respectively.

deep ANNs. While ANN neuron models with continuous functions (such as *sigmoid*, *tanh* or *ReLU*) are compatible with the gradient-based learning, it has been a challenge to directly train SNNs with BP method in their native form. This is due to the spike-output being binary-valued (*i.e.*, zero or one), which renders the spike generation function non-differentiable and discontinuous. To get around this issue, standard BP has been adapted for the spike-based learning domain which we refer to as 'spike-based back-propagation'. The spike-based BP method overcomes the discontinuous spiking functionality by approximately estimating the surrogate gradient of spike generation function. Several surrogate gradient methods have been introduced in the literature [13,14,30]. In this work, we employ the LIF neuronal surrogate gradient function that accounts for the leaky behavior as proposed in [15].

The training procedure is composed of two phases (*e.g.*, forward and backward). In the forward phase, the hidden layer neurons accumulate the weighted sum of spike-inputs in the membrane potential. When this potential exceeds the threshold, the neuron fires a spike-output and resets the potential to the resting state (zero). Otherwise, membrane potential decays exponentially. The final layer neurons do not generate spike output and decay over time, accumulating a weighted sum of spike-inputs. At the last time step, the final prediction outcomes are estimated by dividing the final layer membrane potential ($U_L[T]$) by the total number of time-steps (T). Then, the final errors are evaluated by comparing the final prediction outcomes with the ground truth (*label*). The loss function (*Loss*) is obtained by computing the summation of squared error as shown below,

$$Loss = \frac{1}{2}\left(\frac{U_L[T]}{T} - label\right)^2, \quad \frac{\partial O_l[t]}{\partial U_l[t]} = \frac{1}{V_{th}+\epsilon}(O_l[t] > 0), \tag{6}$$

where $\frac{U_L[T]}{T}$ is the final prediction outcome. In the backward phase, the final errors are propagated backward while unrolling the network in time using the surrogate gradient method. This procedure is often regarded as Backpropagation Through Time (BPTT) [31]. The surrogate gradient of LIF neuronal function is computed by combining the straight through estimation [32] and leak correctional term (ϵ) as given by the second equation in Eq. (6). Here, the straight-through estimation (*i.e.*, $\frac{1}{V_{th}}$) calculates the derivative of IF neuronal function and ϵ compensates the leaky effect of the membrane potential. Finally, the network parameters are updated based on the partial derivatives of the loss with respect to weights for all discrete time steps. The trained SNNs can incorporate temporal and leak statistics from direct spike-inputs over time. The pseudo-code of the spike-based BP learning is given in Algorithm 1.

Algorithm 1. Procedure of spike-based backpropagation learning for an iteration.

Input: pixel-based inputs (*inputs*), total number of time steps (#*timesteps*), number of layers (L), weights (W), membrane potential (U), membrane time constant (τ_m), firing threshold (V_{th})
Initialize: $U_l[t] = 0, \forall l = 1, \ldots, L$
// **Forward Phase**
for $t \leftarrow 1$ **to** #*timesteps* **do**
  // generate Poisson spike-inputs of a mini-batch data
  $O_1[t] = Poisson(inputs)$;
  for $l \leftarrow 2$ **to** $L-1$ **do**
    // membrane potential integrates weighted sum of spike-inputs
    $U_l[t] = U_l[t-1] + W_l O_{l-1}[t]$
    if $U_l[t] > V_{th}$ **then**
      // if membrane potential exceeds V_{th}, a neuron fires a spike
      $O_l[t] = 1, U_l[t] = 0$
    else
      // else, membrane potential decays exponentially
      $O_l[t] = 0, U_l[t] = e^{-\frac{1}{\tau_m}} * U_l[t]$
    end if
  end for
  // final layer neuron does not fire
  $U_L[t] = e^{-\frac{1}{\tau_m}} * U_L[t-1] + W_L O_{L-1}[t]$
end for
// **Backward Phase**
for $t \leftarrow$ #*timesteps* **to** 1 **do**
  for $l \leftarrow L-1$ **to** 1 **do**
    // evaluate partial derivatives of loss with respect to weight by unrolling the network over time
    $\triangle W_l[t] = \frac{\partial Loss}{\partial O_l[t]} \frac{\partial O_l[t]}{\partial U_l[t]} \frac{\partial U_l[t]}{\partial W_l[t]}$
  end for
end for

3. Poisson spike generation under noisy environments

In Section 4.2, the spike-inputs with external random noise are used for experimentally evaluating the noise robustness (*i.e.*, the capability of maintaining a certain prediction accuracy under stochastic perturbations) of multi-layered SNNs. Keeping that goal in mind, here we explain the noisy spike-input generation methods used in our work. Specifically, two different sources of random noise are considered, namely *Gaussian noise* and *Impulse noise* [4]. The choice of these noise sources is mainly inspired from common noise sources that corrupt digital images. During the data acquisition and transmission phase, image quality often deteriorates due to different types of noise and two widely used noise models are Gaussian and Impulse noise [33]. In our work, we train the models with clean (non-noisy) images and test the robustness of LIF and IF models against images which may get noise-corrupted in a real-life scenario, where clean inputs may not always be presented to the neural network. In general, Gaussian noise arises in digital images during acquisition. The variability in the level of illumination and fluctuations in the temperature of the sensor induces some inherent noise in it. Moreover, the electronic circuits inside the sensor may also inject their own circuit noise [34]. This Gaussian noise is often modeled as the most frequently occurring noise in image noise modeling [35], so we also use this as a noise model to investigate our model robustness. Other common noise source is impulse noise which can be caused by analog-to-digital converter errors, bit errors in transmission, etc. [35,36].

Algorithm 2. Poisson spike generation scheme under noise

Input: pixel-based inputs (*inputs*), total number of time steps (#*timesteps*), external random noise (ξ), uniform random number (X)
Output: spike-based inputs ($O_1[t]$)
for $t \leftarrow 1$ **to** #*timesteps* **do**
  if *Scenario*1 **then**
    // External noise (ξ) is added to input pixel
    $inputs_c = inputs + \xi$
    // If noisy input ($inputs_c$) is greater than uniform random number, a spike-input ($O_1[t]$) is generated
    if $inputs_c > X$ **then**
      $O_1[t] = 1$
    else
      $O_1[t] = 0$
    end if
  else if *Scenario*2 **then**
    // External noise (ξ) is added to input channel
    if $inputs > X$ **then**
      $O_1[t] = 1 + \xi$
    else
      $O_1[t] = \xi$
    end if
  end if
end for

We would also like to mention that noisy input currents for LIF neurons are often modeled as Gaussian white noise sources in literature [37,27]. Again, if we consider image processing literature, it is quite common Gaussian and impulse noise sources [38]. Therefore, we consider these two common noise models. For each noise source under consideration, two noise injection scenarios are introduced for producing the noisy spike-inputs. Each noisy spike-input generation procedure is depicted in Algorithm 2. In our analysis, clean spike-inputs refer to homogeneous Poisson spikes where spike-firing probability remains constant in the entire period of input generation. Therefore, inter-spike-intervals (ISI) of such homogeneous Poisson spikes conform to Poisson statistics.

For scenario 1, an independent random noise is added to an image pixel at each time step. The combination of pixel input and noise is compared with an uniformly distributed random number to generate Poisson-distributed spike-inputs. Hence, for a given period of time, the stream of spike-inputs incorporates the noise

over time. In this case, the firing probability varies at every time step, which results in inhomogeneous Poisson spikes with randomly varying firing probability every time step. Here, what varies is the ISI distribution. For scenario 2, an independent random noise is added (at each time step) to the Poisson spike trains generated from the original image pixels. The major difference between two scenarios is whether the random noise is added before or after comparing with a random number (Poisson spike generation process). Note, in scenario 1, spikes are generated as a post-process of adding noise to image pixels, making the input spike train strictly binary, but in scenario 2, noise is added directly to the spikes, so the resultant noisy spikes contain perturbations around their clean spike values (0 or 1). Since the spikes are generated from the Poisson process in scenario 2, the ISI distribution remains identical to that of clean spike-inputs. Instead, the amplitude of each spike varies. The random noise injection process is performed in the input layer only.

4. Experiments

4.1. Experimental setup

We examine the robustness of multi-layered SNNs against noisy spike-inputs on two standard vision benchmarks, namely SVHN and CIFAR-10, which are composed of color (three-dimensional) inputs. We experiment with multi-layered SNN models, which comprise of 32 $\times$ 32 color inputs, convolutional (C) layers with 3 $\times$ 3 weight kernels, average-pooling (P) layers with fixed 2 $\times$ 2 kernel followed by fully-connected (FC) layers. The details of the chosen SNN models are as follows: model used for CIFAR-10 is (32 $\times$ 32-64C3-64C3-2P-128C3-128C3-2P-256C3-256C3-256C3-2s-1024FC-10o) and model used for SVHN is (32 $\times$ 32-64C3-64C3-2P-256C3-256C3-256C3-2s-1024FC-10o). We follow the training protocols as described in [15]. Each network model with different membrane time constant is independently trained with clean training data. Note, the membrane time constant is not considered as a trainable parameter and remains fixed during training and testing. All network models are trained with mini-batch spike-based BP for 150 epochs with a batch size of 64, while decreasing the learning rate at 70^{th} and 100^{th} epoch. After normalizing each image sample to zero mean and scaling to the range $[-1, 1]$, Poisson spike trains are generated for 100 time-steps during training and testing. The reported results are the average score from three independently trained networks. We implemented the multi-layered SNNs using Pytorch deep learning package.

4.2. Robustness against noisy spike-inputs

First, we compare the noise robustness results with different membrane time constants ($e.g., \tau_m = 30, 100$ and *infinity*). The LIF neuron models are associated with relatively smaller membrane time constants ($e.g., \tau_m$ = 30 and 100) compared to IF neuron model with an infinitely large membrane time constant ($e.g., \tau_m$ = *infinity*). Since our goal here is to investigate the effect of leak (determined as $exp(-1/\tau_m)$), we vary τ_m as a hyperparameter among various models. For the IF case, there is no leak, so only possible value of membrane time constant is τ_m = *infinity*. On the other hand, leaky models can have a wide range of τ_m values. On the lower extreme, we observe from our experiments that training loss diverges when the chosen membrane time constant is too small ($\tau_m < 30$). In this case, the spiking activities decrease severely due to extremely high leak while passing through the layers, causing convergence issues in multi-layered SNN training [15]. So, $\tau_m = 30$ is chosen as a high leak model. In addition, we use $\tau_m = 100$ for a low leak model as an in between point between IF and high leak cases. The robustness of each SNN model is measured in terms of the stability of the classification accuracy against noisy spike-inputs. The performances of SNNs are scored across eight severity levels with each noise type (*e.g.*, *Gaussian noise* and *Impulse noise*). The severity level indicates the strength of input noise.

In both benchmark datasets (CIFAR-10, SVHN), the baseline testing accuracy is almost the same under different leak parameters as presented in the first row of Table 1. Fig. 3 shows the accuracy results with increasing level of noise severity across different benchmarks (first row: noisy spike generation scenario 1, second row: noisy spike generation scenario 2). For both the noisy spike generation scenarios, SNNs with LIF neurons (blue, red) achieve improved noise robustness whereas the ones with IF neurons (green) suffer from severe accuracy degradation for high noise severity levels as displayed in Fig. 3. We would like to mention here that all network models are trained on clean spike-inputs, but tested with noisy ones. We adopt such training and testing scheme for two reasons. Firstly, if both models were trained with noisy data in addition to clean data, both models would perform well during testing since the network would be trained to perform well for noisy inputs as well as clean inputs. In that case, it would be difficult to distinguish their performance and we would not be able to investigate the inherent robustness that the leaky behavior offers in LIF models compared to IF ones. Moreover, it would be difficult to decipher whether the resulting robustness is obtained from data augmentation using noisy inputs or from LIF model dynamics. Secondly, the noise that may be present in images during test time is inherently stochastic and thus, it is difficult to account for all different types of noises by training the model with each class of noise. Moreover, models trained with one type of noise may not generalize to other noises. Our goal here is not to enhance model robustness through data augmentation (by incorporating noisy inputs to training data), rather to explore the robustness that the model dynamics provides. So, we adopt the 'train with clean input, and test with noisy ones' approach. Similar scheme has been adopted in [39]. We observe that the models trained with the highest amount of leak (blue) retain the baseline accuracy to a greater extent compared to a non-leaky model. The LIF model with τ_m = 100 shows relatively higher accuracy degradation compared to one with τ_m = 30. However, both models show improved robustness compared to the IF model. These trends hold true for all noisy spike-input generation scenarios.

To further compare between leaky and non-leaky spiking models in presence of noise, we perform an additional experiment. Following the analysis of [17], we train a fully connected network topology with one hidden layer (784-800-10) on MNIST. Two networks are trained with different spike-input cases: the first with non-noisy spike-inputs and the second with noise-corrupted spike-inputs. For noise-corrupted spike input generation, we adopt the noise generation scenario 1 where an independent random Gaussian noise is added to an image pixel at each time step (detailed descriptions are illustrated in Algorithm 2). After training, we validate the classification error with non-noisy spike-inputs and noise-corrupted spike-inputs using the test dataset. Table 2 shows the classification error results for the two different training schemes where in one case, noise is included during training, and in the other case, training is performed without noise. The third row shows the first network result where the network is trained with non-noisy spike-inputs. Similar to the results obtained using SVHN and CIFAR10 dataset, the classification errors of LIF model (e.g., τ_m = 100 and 30) are smaller than IF model when tested with noise-corrupted spike-inputs. Note, the networks with LIF models

Table 1
Comparison between the network models with different leak amounts. The first row corresponds to baseline accuracy. The second and the third rows correspond to the sum-squared errors averaged over 130–150 epochs for testing and training data, respectively. The fourth and the fifth rows correspond to average spiking activity and the total number of synaptic operations, respectively.

Dataset	CIFAR-10			SVHN		
τ_m	30	100	inf	30	100	inf
Accuracy(%)	89.65	90.19	90.30	96.12	96.32	96.32
SSE_{Test}	**2.93**	3.45	3.83	**0.72**	0.75	0.82
SSE_{Train}	**1.88**	1.92	2.20	**1.26**	1.40	1.60
Spikes(%)	**9.45**	5.26	4.94	**14.07**	12.09	11.85
#*SynapticOps*	**1.59E9**	7.92E8	7.18E8	**3.99E9**	3.88E9	3.79E9

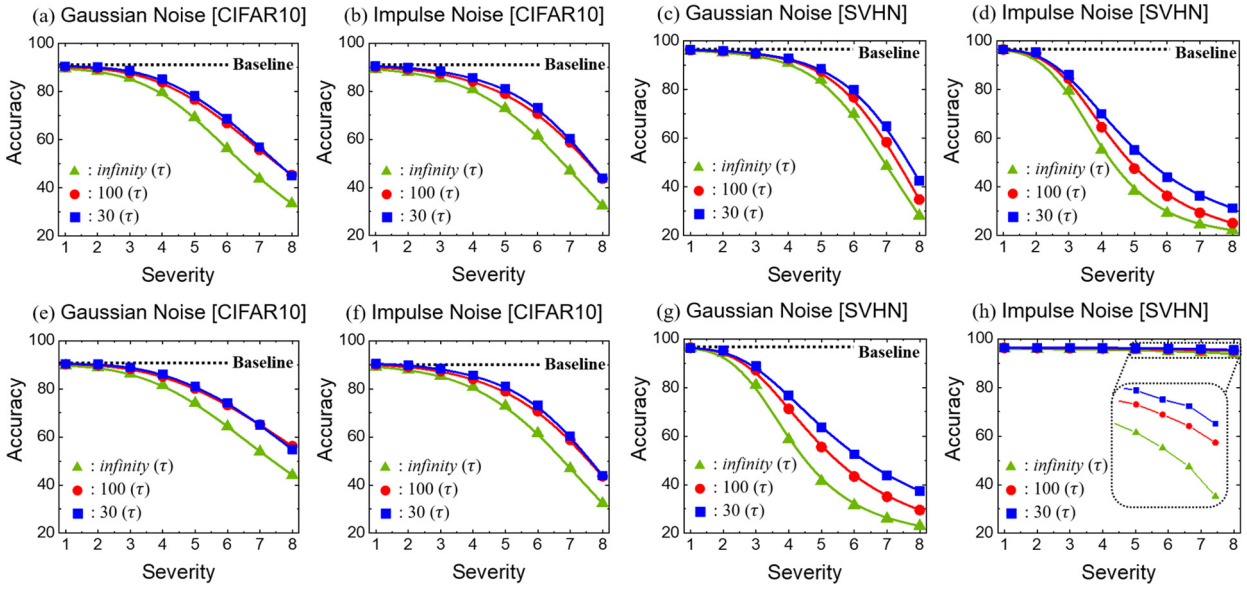


Fig. 3. Classification accuracy at each level of noise severity. The horizontal and vertical axes present the input noise severity and classification accuracy, respectively. (a,b,c,d) Results from noisy input generation scenario 1. (e,f,g,h) Results from noisy input generation scenario 2.

Table 2
Classification error (%) on MNIST.

Train/test scheme	Non-noisy test			Noise-corrupted test		
τ_m	30	100	inf	30	100	inf
Non-noisy training	3.05	2.91	2.99	3.97	4.00	4.05
Noise-corrupted training	3.41	3.47	3.53	3.31	3.36	3.37

(e.g., τ_m = 100 and 30) show higher classification error than the one with IF model when tested with non-noisy spike-inputs, indicating that for clean test data, IF model performs better. Next, the fourth row shows the second network result where the network is trained with noise-corrupted spike-inputs. For this configuration, the network with LIF models show superior test results compared to IF both for noise-corrupted as well as non-noisy test data. To obtain a measure of robustness provided by the models when testing data differs from training data in terms of noise incorporation, we take the difference between classification error of individual models when tested with and without noise. We observe that the network with LIF models (e.g., τ_m = 100 and 30) show smaller classification error difference (0.10% and 0.11%) than the one with IF models (0.16%), which further validates enhanced robustness of LIF models compared to the IF model (see Table 2).

4.3. Spectrum analysis

To analyze the improved noise robustness of LIF models, we perform a spectrum analysis of inputs and the corresponding network outputs for both clean and noisy data. In general, the noise spectrum contains components over a wide frequency band. The single-sided spectra of input-spike trains (averaged over test samples) for the clean and the noisy cases are shown in Fig. 4. It can be observed that the mean spectrum distribution with noisy spike-inputs remains roughly the same compared to the clean case.

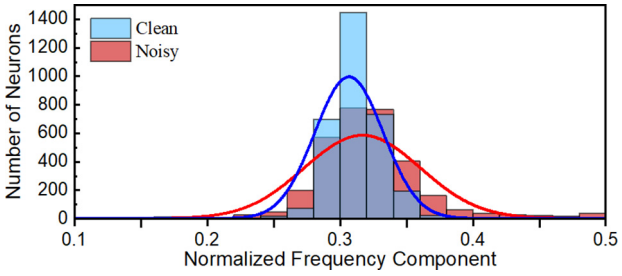


Fig. 4. Histogram of the spectrum of spike trains per image for clean and noisy (Gaussian noise) inputs with corresponding distribution curves.

However, this spectrum of noisy data spreads over a wider band, resulting in more components in higher frequency bands. These changes in the spectrum distribution can significantly alter the spike patterns propagating through the layers compared to the clean input. As previously explained in Section 2.2, the leaky neuron models only pass inputs with low-frequency components. Hence, the leaky neuron models can eliminate some of the high-frequency noise components, thus helping to maintain the baseline performance. However, the low-frequency noise components pass through the LIF and IF models in a similar way. Thus, the accuracy degradation due to such components remains alike for both leaky and non-leaky neurons.

Next, let us consider the spectrum of the target output neuron (node corresponding to the ground truth label) in the final layer, since the changes concerning this output neuron largely determine the correct or wrong classification. For each image, we measure the average spectrum of the target output neuron and calculate the critical frequency up to which the significant power (70%) of the total spectrum resides. This critical frequency distribution is examined over all the samples and plotted in Fig. 5(a)–(c). Interestingly, with an increasing amount of leak, we found that the mean spectrum shifts towards the left. We anticipate this shift towards the lower frequency band is owing to the inherent low-pass filtering effect of leak. The normalized mean critical frequency components for the target neuron corresponding to τ_m = *infinity*, 100 and 30 are 0.345, 0.317 and 0.255 respectively, for the clean testing samples, while for the noisy inputs (for noise severity level of five), the same frequency components become 0.350, 0.330 and 0.298, respectively. These outcomes along with Fig. 5(a)–(c) clearly indicate that frequency components of target neuron's output response become higher with noisy spike-inputs compared to the clean input case. As IF neurons have much wider pass-band, the higher frequency components are not filtered out as shown in Fig. 5(a), thus making the network more prone to have noise errors. In contrast, for the LIF models, most of the high-frequency components are eliminated through the low-pass filtering effect, as demonstrated in Fig. 5(b) and (c) which results in maintaining the baseline accuracy.

4.4. Analyses of generalization

In order to ascertain the improved noise robustness from another perspective, we extend our analysis to generalization. We hypothesize that leaky neuron models enable SNNs to better generalize to previously unseen examples, and examine the impact of leak on generalization. While training multi-layered SNNs over 150 epochs, we recorded the sum of squared error (SSE) on the testing and the training samples to highlight the performance differences. As training progresses towards the end, we found that SNNs with LIF models yield lower testing SSE with the same training effort and reach lower final testing and training SSE than the ones with IF models. The second and the third rows of Table 1 present the testing and training SSE averaged over 130–150 epochs. We also analyze the testing SSE attained as a function of the training SSE. Fig. 5(d,e) shows the testing SSE with respect to the training SSE for different membrane time constants. We found that at the same training SSE, LIF models (blue, red) yield lower testing SSE than IF models (green), hinting towards better generalization. Notably, the advantage of better generalization is the mitigation of overfitting in large neural networks [40,41].

4.5. Input activity analysis

While the enhanced robustness achieved through leaky neuron models is advantageous, it is also pivotal to consider the associated computations and energy costs of using LIF and IF models. To infer an output class, SNNs need the spike-inputs to be fed over a number of times steps, performing event-based synaptic operations that take place only when spike-inputs arrive. In this respect, the total number of synaptic operations is typically considered as a metric for benchmarking the computational costs in neuromorphic hardware [23,42]. This subsection explores the impact of leaky neuron models on the spiking sparsity and the number of computations, two critical factors that directly determine the computational efficiency of SNNs. In Table 1, the fourth and fifth rows present the average spike activities and the total synaptic operations, respectively, for different leak parameters. We found that the overall spiking activities increase with a higher leak, thereby resulting in more synaptic computations. An important insight from here is that, with respect to the degree of leak, there exists a trade-off between noise robustness and compute requirements.

To investigate the reason behind the increased spiking activities with higher leak, we measure the Euclidean norm of the weighted sum of spike-inputs (referred to as 'ENWSI' subsequently) over time for each hidden layer. We would like to note that ENWSI is representative of a combination of spiking activities and weights that determine the net input information to the corresponding layers. Fig. 6 illustrates the ENWSI for different membrane time constants (*e.g.*, τ_m = 30, 100 and *infinity*). We found that SNNs with LIF neurons (blue, red) receive higher ENWSI across the layers than those with IF neurons (green). The model with the largest leak

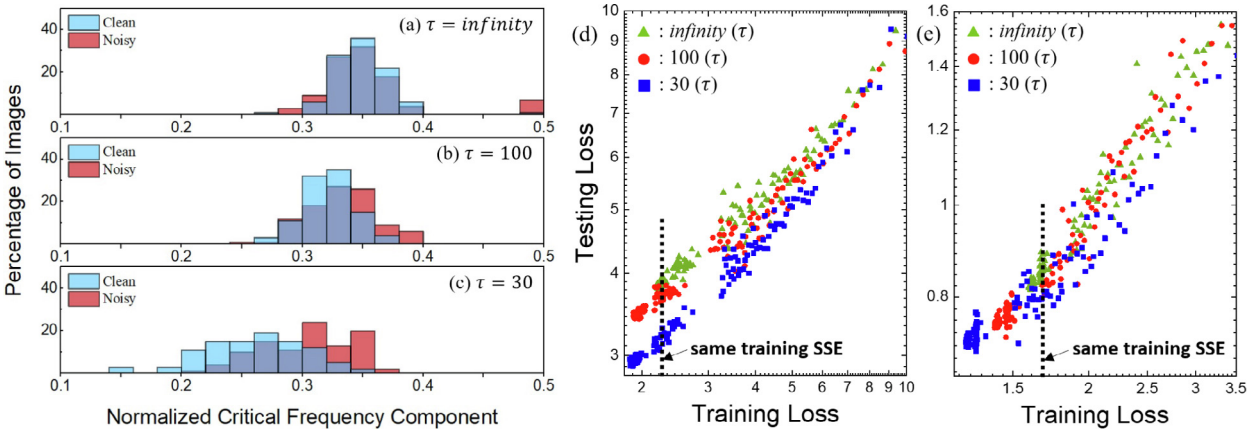


Fig. 5. Histogram of average normalized critical frequency components of target output neuron for (a) τ_m = *infinity*, (b) τ_m = 100 and (c) τ_m = 30. The sum-squared errors of test data with respect to the ones of training data on (d) CIFAR-10 and (e) SVHN benchmarks. The horizontal and vertical axes present the sum-squared error (in log scale) on train and test data, respectively.

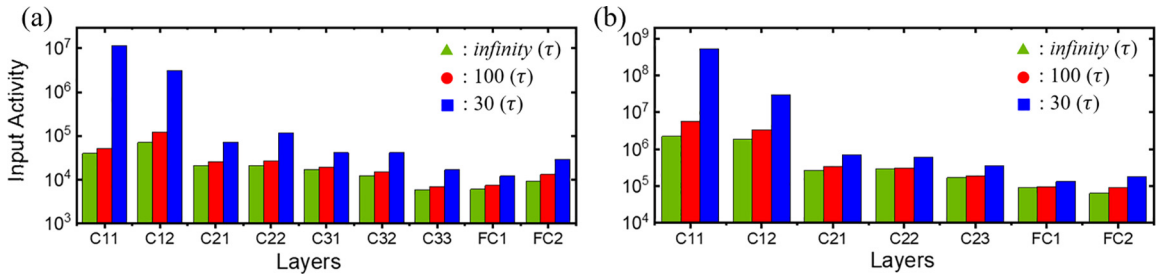


Fig. 6. Layer-wise Euclidean norm of the weighted sum of spike-inputs of multi-layered SNNs for (a) CIFAR-10 and (b) SVHN datasets.

(blue) receives the highest ENWSI compared to the other models under consideration as evidenced in Fig. 6.

It is widely understood that if IF and LIF neurons were to receive the same weighted sum of inputs, the LIF neurons would produce comparatively lesser outputs due to their inherent leak. However, that would lead the LIF models not to have enough spikes in the deeper layers due to the layerwise gradual reduction in spiking activities. Hence, the resultant network would fail to converge with acceptable accuracy. To overcome the leak effect and to have sufficient spiking activities for proper training, spike-based BP training tailors the LIF models to increase the weighted sum of hidden layer input activities beyond what is needed for IF models. Consequently, LIF models converge to configurations with increased spiking activities, allowing for sufficient weighted sum of input activities in the deeper layers. In particular, from Fig. 6, we note that if the ENWSIs are normalized at each layer, the difference in input activity between IF and LIF models is more prominent at the first 2 layers compared to the later layers. Interestingly, if we just consider the ENWSIs for the model with τ_m = 30, we notice an exponential increase if we consider moving from the deeper layers to the initial layers. This is due to the fact that, compared to an IF model, the spiking activity of an LIF model with τ_m = 30 experiences layer-wise exponential decay due to the leak effect. As a result, for the network to learn, it requires much higher ENWSIs at the initial layers so that the deeper layers can have enough spikes to overcome the leak and still propagate sufficient information.

5. Discussion and conclusion

In the neuroscience literature, the existence of leak in biological neurons has been reported in the context of sodium ion channels [20,43], synaptic transmission in visual cortex [44,45], etc. SNN models take the bio-plausibility of leak into account through the leaky neuron models [46,47]. In those models, the leak acts as a hyperparameter that controls the decay of membrane potentials in the neurons over time. However, the effect of leak in learning and the resultant neuronal responses have not been studied comprehensively, to the best of our knowledge. Recognizing this gap, in this paper we investigate IF and LIF neuron models to analyze the role that leak plays in learning and their impact on noise robustness and spiking sparsity. It is to be noted that there are opportunities to further explore the effect of leak, especially in terms of other more complicated neuron models closer to biological ones (*e.g.*, Hodgkin–Huxley [10] and exponential integrate-and-fire [9]). However, till date, there exists a lack of suitable methodologies for training deep SNNs for complex learning tasks with acceptable accuracy using such sophisticated neuron models. Therefore, the study of leak in the context of such models has not been considered in this work.

As regards to studying the consequences of leak, we first focused on the robustness of SNNs to common corruptions, which has been a significant concern in neural networks and motivated a number of recent works. Data augmentation [4,48,49] and quantization [50,51] have been shown to achieve robust performance in both SNNs and ANNs. However, data augmentation-based techniques usually do not generalize well to other types of noise than those used during training, necessitating expensive iterative training efforts using diverse augmented samples. Moreover, the input and weight quantization techniques are reported to be susceptible to error amplification due to enlarged quantization noise in multi-layered networks [52], leading to considerable loss of accuracy. Our work is pertinent in this respect, since the experimental results show that the leaky neuron models enable improved robustness against random noise, without the need for costly re-training procedures or error amplification. We attribute this enhanced robustness to the better generalization and low-pass filtering effects of the LIF neuron models.

However, introducing the leak in the SNN models (while training with backpropagation) comes at the expense of higher spiking activities compared to the IF models. To that effect, with respect to the usage of leaky neuron models, there is a trade-off between noise robustness and computational efficiency. At τ_m = 100, SNNs with LIF neurons achieve substantially improved robustness compared to the ones with IF neurons while maintaining reasonable spiking sparsity. Training with higher leak (τ_m = 30) further improves the robustness; however, the spiking activities also increase considerably. It would be interesting to further validate our findings on large-scale datasets such as Imagenet [53], however it has remained a challenging problem to train SNNs on Imagenet type of datasets satisfactorily without using ANN-SNN conversion techniques. But in those conversion methods, the learning mainly occurs in the ANN domain and hence the effect of leak in the context of SNN domain learning would not be obvious. Since our concentration is not proposing a new learning paradigm, rather exploring the impact of leak parameter in SNN models, we focus on training the SNNs using spike-based backpropagation from scratch and use relatively smaller datasets.

As we analyze the impacts of leak on robustness and sparsity, we expect our study to be particularly useful for designing resource-constrained edge applications in noisy environments (*e.g.*, self-driving vehicles in adverse weather and rescue robots in disasters etc). Considering that leak is an essential bio-plausible element in SNN models, we believe a better understanding of its effects will help to design improved bio-inspired architectures by making optimal choices concerning the involved trade-offs. Furthermore, an efficient algorithm-hardware co-design considering the leak impacts would be of interest for future research directions. To conclude, the understanding of leak provides another knob for designing SNNs, enabling us to obtain a robustly trained network without sacrificing compute-efficiency significantly. Our anticipation is that the findings of this work will contribute towards bridging the two seemingly disparate fields of neuroscience and machine learning.

CRediT authorship contribution statement

Sayeed Shafayet Chowdhury: Methodology, Software, Investigation, Validation, Formal analysis, Writing - original draft, Writing - review & editing. **Chankyu Lee:** Methodology, Software, Investigation, Validation, Formal analysis, Writing - original draft, Writing - review & editing. **Kaushik Roy:** Conceptualization, Resources, Writing - review & editing, Supervision, Project administration, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The work was supported in part by, Center for Brain-inspired Computing (C- BRIC), a DARPA sponsored JUMP center, Semiconductor Research Corporation, National Science Foundation, Intel Corporation, the DoD Vannevar Bush Fellowship and U.S. Army Research Laboratory.

Appendix A. Detailed formulation of coherence function

In this section, we present the derivations of the coherence function, $C(\omega)$ between the input stimulus for the Leaky Integrate and Fire (LIF) neuron and output spike train, guided by [37,54]. The discussion is divided into two parts: first we formulate the equation describing the neuron model in subSection 6.1, next using this formulated model, we derive equations to calculate coherence in subsection A.2.

A.1. LIF model equation

The dynamics of an LIF neuron is modeled as follows:

$$\tau_m \frac{dU}{dt} = -(U - U_{rest}) + RI, \quad U \leqslant V_{th} \tag{A.1}$$

where U is the membrane potential, I denotes the input current, τ_m indicates the membrane time constant, R represents membrane resistance and U_{rest} is the resting potential. Note, an equivalent parallel resistor–capacitor (RC) circuit model of the LIF neuron is illustrated in Fig. 1(c) in the main manuscript.

Let us consider the case where the input $I(t)$ to the model described in Eq. (1) is a white Gaussian noise with a constant mean value $\langle I \rangle$ and a correlation function $\langle (I(t) - \langle I \rangle)(I(t') - \langle I \rangle) \rangle = 2D_I \delta(t - t')$ (here, we denote the mean of a parameter H as $\langle H \rangle$). Let us make the following variable changes:

$$v = \frac{U - U_{rest}}{V_{th} - U_{rest}}, t \longrightarrow \frac{t}{\tau_m}. \tag{A.a}$$

When the membrane potential is U, the input current through the resistance branch of the RC circuit model becomes $(U - U_{rest})/R$. We denote the opposite of this input current as $I_{model}(U) = -(U - U_{rest})/R$. Taking the variable changes from (A.a) into account and differentiating v with respect to time, we get:

$$\frac{dv}{dt} = \dot{v} = \frac{\tau_m}{V_{th} - U_{rest}} \frac{dU}{dt}. \tag{A.b}$$

In addition, using the scaling property of the delta function [26], the correlation function of $I(t)$ becomes $\frac{2D_I}{\tau_m}\delta(t - t')$ (since $\delta(\tau_m t) = \frac{1}{\tau_m}\delta(t)$). Accordingly, we denote input $I(t)$ as follows:

$$I(t) = \langle I \rangle + \sqrt{\frac{2D_I}{\tau_m}}\xi(t), \tag{A.c}$$

where $\xi(t)$ is a zero-mean white Gaussian noise with $\langle \xi(t)\xi(t') \rangle = \delta(t - t')$. Using the relations from (A.b) and (A.c) and dividing both sides by $(V_{th} - U_{rest})$ in Eqn. 1, we obtain the following equation:

$$\dot{v} = -\frac{(U - U_{rest})}{V_{th} - U_{rest}} + \frac{R}{V_{th} - U_{rest}}\langle I \rangle + \frac{R}{V_{th} - U_{rest}}\sqrt{\frac{2D_I}{\tau_m}}\xi(t). \tag{A.2}$$

Considering $v = \frac{U - U_{rest}}{V_{th} - U_{rest}}$, we acquire $(V_{th} - U_{rest})v + U_{rest} = U$. Therefore, we can get:

$$I_{model}((V_{th} - U_{rest})v + U_{rest}) = I_{model}(U) = -\frac{U - U_{rest}}{R} \tag{A.3}$$

Based on Eq. (A.3), the first term on the right-hand side in Eq. (A.2) can be written as:

$$\begin{aligned} \tfrac{R}{V_{th} - U_{rest}} \tfrac{-(U - U_{rest})}{R} &= \tfrac{R}{V_{th} - U_{rest}} [I_{model}((V_{th} - U_{rest})v + U_{rest})] \\ &= \tfrac{R}{V_{th} - U_{rest}} [I_{model}((V_{th} - U_{rest})v + U_{rest}) - I_{model}(U_{rest})] \\ &\quad + \tfrac{R}{V_{th} - U_{rest}} [I_{model}(U_{rest})]. \end{aligned}$$

Next, by merging the time-invariant term, $\frac{R}{V_{th} - U_{rest}}[I_{model}(U_{rest})]$ with the $\frac{R}{V_{th} - U_{rest}}\langle I \rangle$ term on the right-hand side in Eqn. A.2, we can define the $f_{model}(v), \mu$ and D as-

$$\begin{aligned} f_{model}(v) = \frac{R}{V_{th} - U_{rest}} &[I_{model}((V_{th} - U_{rest})v + U_{rest}) \\ &- I_{model}(U_{rest})], \end{aligned} \tag{A.d}$$

$$\mu = \frac{R}{V_{th} - U_{rest}} [\langle I \rangle + I_{model}(U_{rest})], \tag{A.e}$$

and

$$D = \frac{D_I R^2}{\tau_m (V_{th} - U_{rest})^2}. \tag{A.f}$$

Here, μ and D are input parameters that represent the mean and the intensity of the fluctuating input in our model, respectively. Using the definitions from (A.d), (A.e) and (A.f), Eq. (A.2) can be further written as follows:

$$\dot{v} = f_{model}(v) + \mu + \sqrt{2D}\xi(t). \tag{A.4}$$

Now, $I_{model}(U_{rest}) = -(U_{rest} - U_{rest})/R = 0$. Therefore, from Eqn. (A.d), $f_{model}(v)$ for the LIF model can be transformed as follows:

$$\begin{aligned} f_{LIF} = f_{model}(v) &= \tfrac{R}{V_{th} - U_{rest}} [I_{model}((V_{th} - U_{rest})v + U_{rest}) - I_{model}(U_{rest})] \\ &= \tfrac{R}{V_{th} - U_{rest}} [I_{model}((V_{th} - U_{rest})v + U_{rest})]; [\because I_{model}(U_{rest}) = 0] \\ &= \tfrac{R}{V_{th} - U_{rest}} \tfrac{-(U - U_{rest})}{R}; [using\ Eq.\ (3)] \\ &= \tfrac{-(U - U_{rest})}{V_{th} - U_{rest}} = -v; [from (a)], \end{aligned}$$

Therefore, Eq. (A.4) becomes as follows:

$$\dot{v} = -v + \mu + \sqrt{2D}\xi(t), \tag{A.5}$$

which is the formalism also used in [37] and will be followed for the remaining discussions in this study.

A.2. Coherence function

Our analysis is based on the parallel *RC* circuit model of the LIF neuron [24] as depicted by Eq. (1). Here, the membrane capacitance C integrates the input currents over time and the resistance branch R represents the leakage path of membrane potential. For IF

neuron model, since there is no leak path, the R branch is considered as an open circuit. Hence, in this case, $I_{model} = \frac{-(U-U_{rest})}{R} = 0$ and the RC circuit model only contains the capacitor C path. This implies that, for the IF model, R becomes infinity, and correspondingly the membrane time constant τ_m, which is equal to RC, also becomes infinity. On the other hand, for LIF neuron model, the R branch plays a role as the leakage path of membrane potential. When the leakage current through the resistance path increases, the resistance value R and the membrane time constant τ_m gradually decrease. Furthermore, the parameters μ and $D(=\frac{D_I R^2}{\tau_m (V_{th}-U_{rest})^2} = \frac{D_I R}{C(V_{th}-U_{rest})^2})$ become proportional to R according to Eqs. (A.e) and (A.f), respectively. Therefore, for the LIF neuron models, D and μ gradually decrease with the increase in leak amount.

The author in [37] considered $D = D_{bg} + D_{st}$ in Eq. (A.5), where D_{bg} is the background noise intensity, D_{st} is the intensity of the stimulus (Gaussian white noise input) and D is the total noise intensity. For our analysis, by assuming $D_{bg} = 0$, we get $D = D_{st}$ (note, a similar consideration was made in [37] for the results and analysis). Now, let us consider the output spike train of the model described by Eq. (A.5) is $x(t) = \sum \delta[t - t_k]$, where t_k is the k^{th} instant of spike timing, when the input stimulus (s) is Gaussian white noise input. We quantify the information transmission of the spiking model by means of the spectral coherence function. To that end, the Fourier transform of $x(t)$ in a time window $[0, T]$ becomes as follows: $\tilde{x}_T(\omega) = \int_0^T x(t) e^{j\omega t} dt$. The cross-spectrum of output spike train (x) and input stimulus (s) is given as [26]: $S_{x,s}(\omega) = \lim_{T\to\infty} \frac{\langle \tilde{x}(\omega)\tilde{s}^*(\omega)\rangle}{T}$, and the spike train power spectrum is defined as: $S_{x,x}(\omega) = \lim_{T\to\infty} \frac{\langle \tilde{x}(\omega)\tilde{x}^*(\omega)\rangle}{T}$. The coherence function is formally defined as the squared correlation coefficient between the input and output as follows:

$$C_{x,s}(\omega) = \frac{|S_{x,s}(\omega)|^2}{S_{x,x}(\omega) S_{s,s}(\omega)}. \tag{A.6}$$

The coherence function $C_{x,s}(\omega)$ generates an output number between 0 and 1 at each measurement frequency. The amount of information transmission at each frequency is proportional to the coherence at that particular frequency, with 1 and 0 denoting full and null transmission, respectively. For a system acting as a low-pass filter, the coherence output under white-noise stimulation decreases in the high-frequency domain.

Next, we analyze the low-pass filtering effect of the LIF neuron model as described by Eq. (A.5). The analytical expression for $S_{x,s}(\omega)$ is given as follows [37,55,56]:

$$S_{x,s}(\omega) = \frac{2D_{st}}{\sqrt{D}} \frac{r_0 i\omega}{i\omega - 1} \frac{\mathscr{D}_{i\omega-1}\left(\frac{\mu-V_{th}}{\sqrt{D}}\right) - e^{\Delta}\mathscr{D}_{i\omega-1}\left(\frac{\mu-U_{rest}}{\sqrt{D}}\right)}{\mathscr{D}_{i\omega}\left(\frac{\mu-V_{th}}{\sqrt{D}}\right) - e^{\Delta} e^{i\omega\tau_r}\mathscr{D}_{i\omega}\left(\frac{\mu-U_{rest}}{\sqrt{D}}\right)}, \tag{A.7}$$

where $\Delta = \frac{U_{rest}^2 - V_{th}^2 + 2\mu(V_{th}-U_{rest})}{4D}$, τ_r is the refractory period and $\mathscr{D}(x)$ is the parabolic cylinder function. In our case, we follow the same assumptions as in [37] where $U_{rest} = 0$, $\tau_r = 0$ and $V_{th} = 1$. The firing rate r_0 is given by calculating the following [37]:

$$r_0 = \left[\tau_r + \sqrt{\pi}\int_{\frac{\mu-V_{th}}{\sqrt{2D}}}^{\frac{\mu-U_{rest}}{\sqrt{2D}}} dz e^{z^2} erfc(z)\right]^{-1}.$$

The power spectrum of the output spike train is given by [57], calculated as follows:

$$S_{x,x}(\omega) = r_0 \frac{\left|\mathscr{D}_{i\omega}\left(\frac{\mu-V_{th}}{\sqrt{D}}\right)\right|^2 - e^{2\Delta}\left|\mathscr{D}_{i\omega}\left(\frac{\mu-U_{rest}}{\sqrt{D}}\right)\right|^2}{\left|\mathscr{D}_{i\omega}\left(\frac{\mu-V_{th}}{\sqrt{D}}\right) - e^{\Delta} e^{i\omega\tau_r}\mathscr{D}_{i\omega}\left(\frac{\mu-U_{rest}}{\sqrt{D}}\right)\right|^2}, \tag{A.8}$$

and the noise input spectrum becomes [37]:

$$S_{s,s}(\omega) = 2D_{st}. \tag{A.9}$$

Taking the magnitude square of the quantity in Eq. (A.7), we derive the following,

$$|S_{x,s}(\omega)|^2 = \frac{4D_{st}^2}{D} \frac{r_0^2\omega^2}{1+\omega^2} \frac{|\mathscr{D}_{i\omega-1}\left(\frac{\mu-V_{th}}{\sqrt{D}}\right) - e^{\Delta}\mathscr{D}_{i\omega-1}\left(\frac{\mu-U_{rest}}{\sqrt{D}}\right)|^2}{|\mathscr{D}_{i\omega}\left(\frac{\mu-V_{th}}{\sqrt{D}}\right) - e^{\Delta} e^{i\omega\tau_r}\mathscr{D}_{i\omega}\left(\frac{\mu-U_{rest}}{\sqrt{D}}\right)|^2}. \tag{A.10}$$

Finally, plugging the values of $S_{x,x}(\omega), S_{s,s}(\omega)$ and $|S_{x,s}(\omega)|^2$ into Eq. (A.6), we obtain the resultant coherence function as follows:

$$C_{x,s}(\omega) = \frac{2D_{st}}{D} \frac{r_0\omega^2}{1+\omega^2} \frac{\left|\mathscr{D}_{i\omega-1}\left(\frac{\mu-V_{th}}{\sqrt{D}}\right) - e^{\Delta}\mathscr{D}_{i\omega-1}\left(\frac{\mu-U_{rest}}{\sqrt{D}}\right)\right|^2}{\left|\mathscr{D}_{i\omega}\left(\frac{\mu-V_{th}}{\sqrt{D}}\right)\right|^2 - e^{2\Delta}\left|\mathscr{D}_{i\omega}\left(\frac{\mu-U_{rest}}{\sqrt{D}}\right)\right|^2}. \tag{A.11}$$

References

[1] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.

[2] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of go with deep neural networks and tree search, Nature 529 (7587) (2016) 484–489.

[3] D. Li, X. Chen, M. Becchi, Z. Zong, Evaluating the energy efficiency of deep convolutional neural networks on cpus and gpus, in: 2016 IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom) (BDCloud-SocialCom-SustainCom), IEEE, 2016, pp. 477–484.

[4] D. Hendrycks, T. Dietterich, Benchmarking neural network robustness to common corruptions and perturbations, in: International Conference on Learning Representations, 2018.

[5] K. Roy, A. Jaiswal, P. Panda, Towards spike-based machine intelligence with neuromorphic computing, Nature 575 (7784) (2019) 607–617.

[6] W. Maass, Networks of spiking neurons: the third generation of neural network models, Neural Networks 10 (9) (1997) 1659–1671.

[7] A.V. Herz, T. Gollisch, C.K. Machens, D. Jaeger, Modeling single-neuron dynamics and computations: a balance of detail and abstraction, Science 314 (5796) (2006) 80–85.

[8] R. FitzHugh, Impulses and physiological states in theoretical models of nerve membrane, Biophysical Journal 1 (6) (1961) 445.

[9] R. Brette, W. Gerstner, Adaptive exponential integrate-and-fire model as an effective description of neuronal activity, Journal of Neurophysiology 94 (5) (2005) 3637–3642.

[10] A.L. Hodgkin, A.F. Huxley, Currents carried by sodium and potassium ions through the membrane of the giant axon of loligo, The Journal of Physiology 116 (4) (1952) 449–472.

[11] E.M. Izhikevich, Simple model of spiking neurons, IEEE Transactions on Neural Networks 14 (6) (2003) 1569–1572.

[12] C. Teeter, R. Iyer, V. Menon, N. Gouwens, D. Feng, J. Berg, A. Szafer, N. Cain, H. Zeng, M. Hawrylycz, et al., Generalized leaky integrate-and-fire models classify multiple neuron types, Nature Communications 9 (1) (2018) 1–15.

[13] J.H. Lee, T. Delbruck, M. Pfeiffer, Training deep spiking neural networks using backpropagation, Frontiers in Neuroscience 10 (2016) 508.

[14] D. Huh, T.J. Sejnowski, Gradient descent for spiking neural networks, in: Advances in Neural Information Processing Systems, 2018, pp. 1433–1443.

[15] C. Lee, S.S. Sarwar, P. Panda, G. Srinivasan, K. Roy, Enabling spike-based backpropagation for training deep neural network architectures, Frontiers in Neuroscience (2020)..

[16] P. Dayan, L.F. Abbott, Theoretical Neuroscience, vol. 806, MIT Press, Cambridge, MA, 2001.

[17] H. Mostafa, Supervised learning based on temporal coding in spiking neural networks, IEEE Transactions on Neural Networks and Learning Systems 29 (7) (2017) 3227–3235.

[18] S.R. Kheradpisheh, T. Masquelier, Temporal backpropagation for spiking neural networks with one spike per neuron, International Journal of Neural Systems 30 (06) (2020) 2050027.

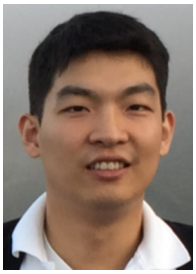
[19] M. Zhang, J. Wang, B. Amornpaisannon, Z. Zhang, V. Miriyala, A. Belatreche, H. Qu, J. Wu, Y. Chua, T.E. Carlson, et al., Rectified linear postsynaptic potential function for backpropagation in deep spiking neural networks, arXiv preprint arXiv:2003.11837, 2020..

[20] T.P. Snutch, A. Monteil, The sodium "leak" has finally been plugged, Neuron 54 (4) (2007) 505–507.

[21] A.D. Rast, F. Galluppi, X. Jin, S.B. Furber, The leaky integrate-and-fire neuron: A platform for synaptic model exploration on the spinnaker chip, in: The 2010 International Joint Conference on Neural Networks (IJCNN), IEEE, 2010, pp. 1–8.

[22] P.A. Merolla, J.V. Arthur, R. Alvarez-Icaza, A.S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, et al., A million spiking-neuron integrated circuit with a scalable communication network and interface, Science 345 (6197) (2014) 668–673.
[23] M. Davies, N. Srinivasa, T.-H. Lin, G. Chinya, Y. Cao, S.H. Choday, G. Dimou, P. Joshi, N. Imam, S. Jain, et al., Loihi: A neuromorphic manycore processor with on-chip learning, IEEE Micro 38 (1) (2018) 82–99.
[24] W. Gerstner, W.M. Kistler, R. Naud, L. Paninski, Neuronal Dynamics, Cambridge University Press, 2014.
[25] N. Fourcaud-Trocmé, D. Hansel, C. Van Vreeswijk, N. Brunel, How spike generation mechanisms determine the neuronal response to fluctuating inputs, Journal of Neuroscience 23 (37) (2003) 11628–11640.
[26] J.G. Proakis, Digital Signal Processing: Principles Algorithms and Applications, Pearson Education India, 2001.
[27] N. Sharafi, J. Benda, B. Lindner, Information filtering by synchronous spikes in a neural population, Journal of Computational Neuroscience 34 (2) (2013) 285–301.
[28] W.M. Connelly, M. Laing, A.C. Errington, V. Crunelli, The thalamus as a low pass filter: filtering at the cellular level does not equate with filtering at the network level, Frontiers in Neural Circuits 9 (2016) 89.
[29] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985..
[30] N. Rathi, G. Srinivasan, P. Panda, K. Roy, Enabling deep spiking neural networks with hybrid conversion and spike timing dependent backpropagation, in: International Conference on Learning Representations, 2019.
[31] P.J. Werbos, Backpropagation through time: what it does and how to do it, Proceedings of the IEEE 78 (10) (1990) 1550–1560.
[32] Y. Bengio, N. Léonard, A. Courville, Estimating or propagating gradients through stochastic neurons for conditional computation, arXiv preprint arXiv:1308.3432, 2013..
[33] R.C. Gonzalez, R.E. Woods, et al., Digital image processing (2002).
[34] D.P. Cattin, "Image restoration: Introduction to signal and image processing," MIAC, University of Basel. Retrieved 11 (2013) 93.
[35] C. Boncelet, Image noise models, in: The Essential Guide to Image Processing, Elsevier, 2009, pp. 143–167.
[36] L.G. Shapiro, G. Stockman, Computer Vision, Prentice Hall Inc., New Jersey, 2001.
[37] B. Lindner, Low-pass filtering of information in the leaky integrate-and-fire neuron driven by white noise, in: International Conference on Theory and Application in Nonlinear Dynamics (ICAND 2012), Springer, 2014, pp. 249–258..
[38] A. Awad, Denoising images corrupted with impulse, gaussian, or a mixture of impulse and gaussian noise, Engineering Science and Technology, an International Journal 22 (3) (2019) 746–753.
[39] M. Zhang, H. Qu, X. Xie, J. Kurths, Supervised learning in spiking neural networks with noise-threshold, Neurocomputing 219 (2017) 333–349.
[40] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, S. Bengio, Why does unsupervised pre-training help deep learning?, Journal of Machine Learning Research 11 (Feb) (2010) 625–660
[41] C. Lee, P. Panda, G. Srinivasan, K. Roy, Training deep spiking convolutional neural networks with stdp-based unsupervised pre-training followed by supervised fine-tuning, Frontiers in Neuroscience 12 (2018) 435.
[42] P.A. Merolla, J.V. Arthur, R. Alvarez-Icaza, A.S. Cassidy, J. Sawada, F. Akopyan, B. L. Jackson, N. Imam, C. Guo, Y. Nakamura, et al., A million spiking-neuron integrated circuit with a scalable communication network and interface, Science 345 (6197) (2014) 668–673.
[43] D. Ren, Sodium leak channels in neuronal excitability and rhythmic behaviors, Neuron 72 (6) (2011) 899–911.
[44] Ö.B. Artun, H.Z. Shouval, L.N. Cooper, The effect of dynamic synapses on spatiotemporal receptive fields in visual cortex, in: Proceedings of the National Academy of Sciences, 1998, pp. 11999–12003.
[45] D. Millman, S. Mihalas, A. Kirkwood, E. Niebur, Self-organized criticality occurs in non-conservative neuronal networks during 'up'states, Nature Physics 6 (10) (2010) 801–805.
[46] N. Rathi, K. Roy, Diet-snn: Direct input encoding with leakage and threshold optimization in deep spiking neural networks, arXiv preprint arXiv:2008.03658, 2020..
[47] I. Garg, S.S. Chowdhury, K. Roy, Dct-snn: Using dct to distribute spatial information over time for learning low-latency spiking neural networks, arXiv preprint arXiv:2010.01795, 2020..
[48] E. Hunsberger, C. Eliasmith, Spiking deep networks with LIF neurons, 2015, arXiv preprint arXiv:1510.08829.
[49] I. Vasiljevic, A. Chakrabarti, G. Shakhnarovich, Examining the impact of blur on recognition by convolutional networks, arXiv preprint arXiv:1611.05760, 2016..
[50] S. Han, H. Mao, W.J. Dally, Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding, arXiv preprint arXiv:1510.00149, 2015..
[51] P. Panda, I. Chakraborty, K. Roy, Discretization based solutions for secure machine learning against adversarial attacks, IEEE Access 7 (2019) 70157–70168.
[52] J. Lin, C. Gan, S. Han, Defensive quantization: When efficiency meets robustness, 2019, arXiv preprint arXiv:1904.08444.
[53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.
[54] R.D. Vilela, B. Lindner, Are the input parameters of white noise driven integrate and fire neurons uniquely determined by rate and cv?, Journal of Theoretical Biology 257 (1) (2009) 90–99
[55] B. Lindner, L. Schimansky-Geier, Transmission of noise coded versus additive signals through a neuronal ensemble, Physical Review Letters 86 (14) (2001) 2934.
[56] B. Lindner, J. Garcia-Ojalvo, A. Neiman, L. Schimansky-Geier, Effects of noise in excitable systems, Physics Reports 392 (6) (2004) 321–424.
[57] B. Lindner, L. Schimansky-Geier, A. Longtin, Maximizing spike train coherence or incoherence in the leaky integrate-and-fire model, Physical Review E 66 (3) (2002) 031916.

Sayeed Shafayet Chowdhury received his B.Sc. degree in Electrical & Electronic Engineering from Bangladesh University of Engineering and Technology in 2016. He won the IEEE Signal Processing Cup (SP Cup) competition in 2015 and 2016, also obtained 3rd position in the IEEE Video & Image Processing Cup (VIP Cup) competition in 2017. Currently, he is a 3rd year Ph.D. student in the School of Electrical and Computer Engineering at Purdue University, West Lafayette, IN, under the guidance of Prof. Kaushik Roy. His current research interests include neuromorphic computing, specifically, developing energy-efficient algorithms for deep spiking neural networks (recognition, inference, analytics), event-based vision and embodied AI.



Chankyu Lee is a Ph.D. candidate in the School of Electrical and Computer Engineering at Purdue University, West Lafayette, IN. He received his B.S. in Electrical and Electronics Engineering from Sungkyunkwan University (SKKU), South Korea, in 2015. He conducted research on brainmachine interface at the Advanced Institutes of Convergence Technology, Republic of Korea, in the summer 2014. In addition, he worked as a graduate intern at Nokia Bell Labs, Murray Hill NJ, in summer 2018. He has been focused on developing energy-efficient/robust deep learning algorithms, with special interests in spiking neural networks, computer vision for event-based cameras, and low-power/high-performance VLSI design for deep learning hardware.

Kaushik Roy received B.Tech. degree in electronics and electrical communications engineering from the Indian Institute of Technology, Kharagpur, India, and Ph.D. degree from the electrical and computer engineering department of the University of Illinois at Urbana-Champaign in 1990. He was with the Semiconductor Process and Design Center of Texas Instruments, Dallas, where he worked on FPGA architecture development and low-power circuit design. He joined the electrical and computer engineering faculty at Purdue University, West Lafayette, IN, in 1993, where he is currently Edward G. Tiedemann Jr. Distinguished Professor. He also the director of the center for brain-inspired computing (C-BRIC) funded by SRC/DARPA. His research interests include neuromorphic and emerging computing models, neuro-mimetic devices, spintronics, device-circuit-algorithm co-design for nano-scale Silicon and non-Silicon technologies, and lowpower electronics. Dr. Roy has published more than 800 papers in refereed journals and conferences, holds 28 patents, supervised 91 PhD dissertations, and is co-author of two books on Low Power CMOS VLSI Design (John Wiley & McGraw Hill).

Dr. Roy received the National Science Foundation Career Development Award in 1995, IBM faculty partnership award, ATT/Lucent Foundation award, 2005 SRC Technical Excellence Award, SRC Inventors Award, Purdue College of Engineering Research Excellence Award, Outstanding Mentor Award in 2021, Humboldt Research Award in 2010, 2010 IEEE Circuits and Systems Society Technical Achievement Award (Charles Desoer Award), IEEE TCVLSI Distinguished Research Award in 2021, Distinguished Alumnus Award from Indian Institute of

Technology (IIT), Kharagpur, Fulbright-Nehru Distinguished Chair, DoD Vannevar Bush Faculty Fellow (2014-2019), Semiconductor Research Corporation Aristotle award in 2015, and best paper awards at 1997 International Test Conference, IEEE 2000 International Symposium on Quality of IC Design, 2003 IEEE Latin American Test Workshop, 2003 IEEE Nano, 2004 IEEE International Conference on Computer Design, 2006 IEEE/ACM International Symposium on Low Power Electronics & Design, 2005 and 2019 IEEE Circuits and system society Outstanding Young Author Award (Chris Kim, Abhronil Sengupta), 2006 IEEE Transactions on VLSI Systems best paper award, 2012 ACM/IEEE International Symposium on Low Power Electronics and Design best paper award, 2013 IEEE Transactions on VLSI Best paper award. Dr. Roy was a Purdue University Faculty Scholar (1998–2003). He was a Research Visionary Board Member of Motorola Labs (2002) and held the M. Gandhi Distinguished Visiting faculty at Indian Institute of Technology (Bombay) and Global Foundries visiting Chair at National University of Singapore. He has been in the editorial board of IEEE Design and Test, IEEE Transactions on Circuits and Systems, IEEE Transactions on VLSI Systems, and IEEE Transactions on Electron Devices. He was Guest Editor for Special Issue on Low-Power VLSI in the IEEE Design and Test (1994) and IEEE Transactions on VLSI Systems (June 2000), IEE Proceedings – Computers and Digital Techniques (July 2002), and IEEE Journal on Emerging and Selected Topics in Circuits and Systems (2011). Dr. Roy is a fellow of IEEE.