Efficient Optimized Spike Encoding for Spiking Neural Networks

Dighanchal Banerjee DIGHANCHAL.B@TCS.COM
Sounak Dey SOUNAK.D@TCS.COM
Arun M. George ARUNM.GEORGE@TCS.COM
Arijit Mukherjee MUKHERJEE.ARIJIT@TCS.COM
TCS Research & Innovation, Tata Consultancy Services, India

Abstract

In recent years, spiking neural networks (SNN) have evolved as a means of energy & data efficient edge computing candidate. However, to achieve better performance from spiking neuron models, the real valued inputs that they process must be efficiently encoded into spike trains. This paper proposes an information theoretic optimization approach by maximizing mutual information between a sequence of real valued data and corresponding encoded spike trains, which in turn increases the efficiency of an SNN-based reservoir up to 14% while performing reconstruction of time series as tested on four different data sets.

1. Introduction

The extreme power efficiency of data processing and decision making in mammalian brains is largely due to adaptive algorithms and techniques used at each stage of cortical process flow [4], an example being how brain parses and encodes any external stimuli into an optimal format for further processing. While cortical architecture and algorithms are not fully understood yet, different mathematical models of spiking neurons are being applied efficiently for performing various tasks like image classification, time-series forecasting, gesture recognition from video [2, 9]. SNNs perform more efficiently if the captured external stimuli is properly encoded while retaining maximum possible information content. Two currently used methods are - (i) capturing the data in an efficient SNN-compatible manner using modern sensor hardware, and, (ii) using efficient soft encoding techniques on the captured data. Dynamic Vision Sensor (DVS) [1] follows the first paradigm by recording scenes like a human eye such that only pixel-level changes in luminosity between consecutive frames are recorded, thus reducing data redundancy and contributing to the efficiency of spiking networks. Unfortunately, not all real world stimuli can be recorded in such a fashion, and for real-valued data, the second option is the only applicable one.

Two popular soft encoding techniques exist in the SNN paradigm: *rate coding*, based on neuron firing rate, and *temporal coding*, based on the timing of the spike. During encoding, *deterministic* and *probabilistic* measures can be used for optimizing the encoded spike train. *Deterministic* measures (such as bin metric, convolution metric etc.) of encoding work better with static data like images, whereas *probabilistic* measures (such as serial correlation, entropy, mutual information etc.) work well for temporally varying data (time series) [5, 10]. Typically, a real valued data stream is encoded into spike train using probabilistic distributions like Poisson, Bernoulli etc. (henceforth called *"base encoding"*) but this does not ensure retention of maximum information.

1. https://inivation.com/wp-content/uploads/2019/08/DVS128.pdf

© D. Banerjee, S. Dey, A.M. George & A. Mukherjee.

Contributions: We propose an information theoretic optimization approach where mutual information (MI) between entire sequence of real valued stimuli and corresponding encoded spike trains (w.r.t. base encoding) is maximized by introducing an optimal Gaussian noise that augments the entire original data. This offers two advantages: (i) reduced loss of relevant information during encoding process resulting in enhanced performance of the spiking network, and, (ii) applicability on temporally varying data as mutual information is calculated across time length of the spike train. We have proved our concept by reconstructing few well known time series using a reservoir based spiking neural network, and we found that with our encoding scheme, 4-14% more accuracy can be obtained during reconstruction of the time series compared to base encoding.

2. Related works

The spike encoders that are commonly used in SNN domain can be categorised into two broad genres: *with decoder* and *without decoder*.

Temporal-Contrast (TC) [16], Hough Spiker Algorithm (HSA) [13] and Bens Spiker Algorithm (BSA) [22] etc. are examples of the former type of algorithms. The decoders associated with these help in comparing the output of a spiking network with the desired output. A predetermined threshold value or filter parameters independent of the nature and size of the data is used in these algorithms.

For the second genre of algorithms, retention of data features in the encoded spike train is obtained using deterministic and probabilistic measures. Bruno et al. [6] proposed a few deterministic measures while [5, 10] proved robustness of probabilistic measures in case of temporally varying data. Alexander et al. [3] validated information theory to build simple stimulus–response models for neural coding and Perkel et al. [20] used statistical measures for inter-spike intervals on static stimuli. Algorithms for approximation of probabilistic inference is categorized based on parametric variation and sampling. Combining these two approaches, Shivkumar et al. [23] portrayed the neural responses as samples from the probabilistic model previously learnt for given inputs, as well as the parameters of the succeeding distribution of the data. But using these stochastic codes may trigger loss of information. Petro et al. [21] worked on error optimization between the original and decoded signal for temporal coding algorithms. Taherkhani et al. [24] worked towards optimizing output spike train by adjusting weights of a spiking neural network in a supervised manner. In our work, instead of optimizing the network parameters, we adjust the input spike train by optimizing the mutual information, a probabilistic measure that gives the idea of the best possible representation of the data. We encourage the readers to refer to Appendix A for further details.

3. Encoding Optimization Approach & Application

Optimization: A standard practice in SNN application domain is to encode real valued data into spike trains "base encoding" (a detailed account of encoding mechanism is given in Appendix B). Gustaf et. al, in [7], shows how entropy (and in turn MI) is calculated by using "probability of spiking rate given the stimulus value". In our approach, we have considered the entire spike train as that carries richer temporal information than rate of spike alone. The mutual information between two quantities is calculated in terms of joint *probability mass function* (PMF) and marginal PMF of those two quantities (refer to Appendix C.1). We have used the same approach to calculate MI between data and encoded spike train. To calculate the probability of a spike train given the real valued signal data, we introduced a small Gaussian noise into the data to bring in the variability.

Let S be the original stimulus and T be the corresponding spike train created with Poisson (base) encoding. Also let T_σ be the encoded spike train for the signal $S + \sigma_S$, where σ_S is the Gaussian noise. Then the optimization problem can be formulated as:

$$\begin{aligned} \underset{\sigma}{\text{maximize}} \quad & I(T_\sigma; S) \\ \text{subject to} \quad & I(T_\sigma; S) >= I(T; S) \end{aligned} \tag{1}$$

The encoded spike train, for which MI is found be maximum, is chosen as the optimized one. This spike train, obtained against a particular optimal noise, is hypothesized *to help a spiking network perform better than that using base encoded spike train.*

Application: In order to test the validity of above hypothesis, the optimized spike train containing maximum information about the input signal is fed into a reservoir of spiking neurons. The reservoir is supposed to learn the temporal dynamics of the encoded spike train. A detailed description of neuron models and reservoir is given in Appendix C. The post-synaptic trace of a spiking neural network, being the representative of the memory of spiking activity inside the reservoir, is then fed into a Linear Regression module with corresponding real values from the input signal as labels to learn (during training) and then to reconstruct the data from the unseen parts of it during validation. Unlike temporal encoding algorithms such as HSA, BSA etc., this Linear Regression module learns a map between synaptic trace of reservoir to the real signal values. The workflow is described in Fig. 1.

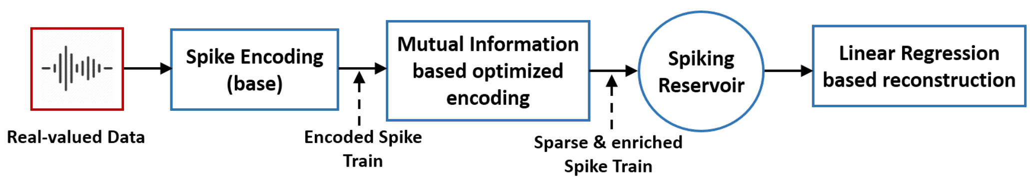


Figure 1: Proposed workflow for optimized encoding

4. Datasets, Experiments and Results

Dataset & Setup: The hypothesis has been tested on four time series data. The first one, namely Mackey-Glass Time series (MGS) [19], is a non-linear chaotic time-series generated by the delay-differential equation:

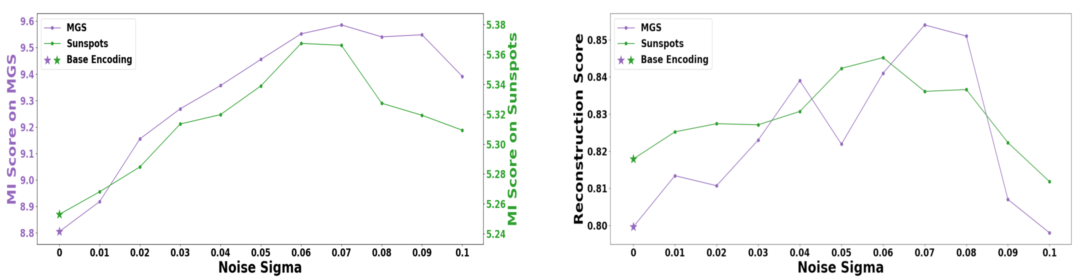
$$\tau \frac{dx}{dt} = \beta x(t) + \alpha \frac{x(t-\tau)}{(1 + x(t-\tau)^{10})} \tag{2}$$

Many biological process dynamics can be modelled using Equation 2. By varying the value of scale constants (α, β) and time delay (τ), one can obtain different MGS time series. Here we have taken α = 2 , β = -1 and τ = 2. The three other time series data that we used were: (i) *Shampoo Sales* [25] - the monthly count of sales of shampoo over a three year period (36 observations), (ii) *Sun Spot* [1] - a monthly count of observed sunspots for from 1749 to 1983 (2820 observations), and, (iii) *UCI Bike Sharing* [8] - hourly count of bike rentals for two years (2011-12) in Capital bike share system (17800 observations). *The purpose of taking data sets with varying number of observations is to test its effect on our proposed method.*

The spiking neural network is implemented using *BindsNet 0.2.7* [11], a GPU based open source SNN simulator in Python. Details of the neuron and reservoir network parameters are provided in the Appendix D.

The reconstruction efficiency of the network is measured using *Coefficient of determination* (R^2 Score) metric (refer Appendix E). It is referred as ***reconstruction score*** in the paper.

Experiments & Results: Each dataset was partitioned into training and testing sets and encoded by maximizing mutual information. For each case, the encoded training set was used to train a spiking reservoir network to reconstruct the time series and then the reconstruction score was measured using the testing set. We explain the procedure w.r.t. MGS. This time series is divided into two parts with training & testing data ratio being 3:2. The purple line in Fig. 2(*a*) shows how the mutual information score varies with respect to different Gaussian noise (σ). As can be seen from the figure, the quantity mutual information increases as the input noise increases and decreases after a certain noise level (σ = 0.07, in this case) making the whole signal too noisy and unusable. So, this inflection point corresponding to σ = 0.07 can be considered as the optimized mutual information available between the input data and the encoded spike train. The coorresponding spike train is considered to be the optimized one which carries maximum information about MGS time series data.



(*a*) Mutual information score

(*b*) Reconstruction score

Figure 2: Variation of mutual information and reconstruction score with increasing noise for MGS (purple) & Sun Spot (green) dataset

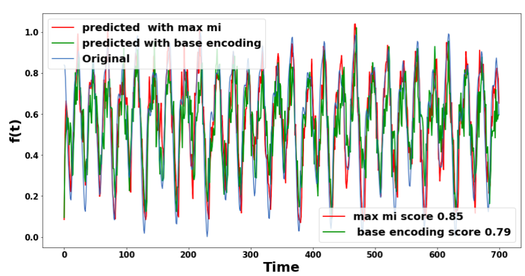
The effect of increasing noise (and in turn changing MI) on the learning capability of spiking reservoir and the Linear Regression module can be observed in terms of reconstruction score for the validation part of the time series data. The effect of increasing noise on reconstruction score (for MGS data) is reported in Fig. 2(*b*) (purple line). It is to be noted that the reconstruction performance of the spiking network improves when compared to base encoding with the increase of noise. Maximum reconstruction score of 0.85 is achieved for the highest mutual information score (9.58) for σ = 0.07. After reaching the optimum point, the reconstruction score decreases due to high noise and eventually comes below the performance of the base encoding. We conducted further experiments on MGS data with different training & testing ratio to observe the effect of varying training data size on reconstruction score. In each case, the spike train corresponding to maximum MI is taken and the train:test data configuration along with corresponding reconstruction score is reported in second column of Table 1.

We performed similar experiments on the three remaining time series datasets and found similar behaviour between MI and noise in all three cases as it was in the case of MGS data. The green lines in Fig. 2(*a*) and Fig. 2(*b*) shows the change in MI and reconstruction score respectively for the Sun

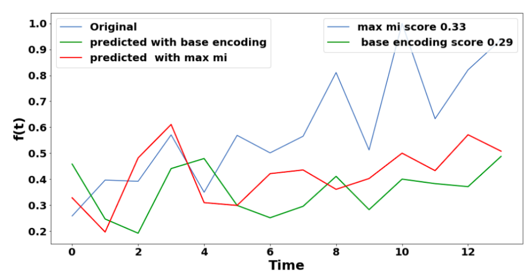
Train & Test ratio	Reconstruction Score							
	MGS		Shampoo Sales		Sunspot		Bike Sharing	
	B	MI	B	MI	B	MI	B	MI
1:4	0.76	0.78	0.21	0.21	0.80	0.81	0.46	0.47
2:3	0.78	0.81	0.26	0.28	0.82	0.83	0.48	0.49
3:2	**0.79**	**0.85**	**0.29**	**0.33**	**0.81**	**0.84**	0.50	0.52
4:1	0.75	0.75	0.29	0.30	0.81	0.81	**0.48**	**0.53**

Table 1: Reconstruction score of four time series data set for different train:test ratios (here, B = Base encoding, MI = Encoding with mutual information)

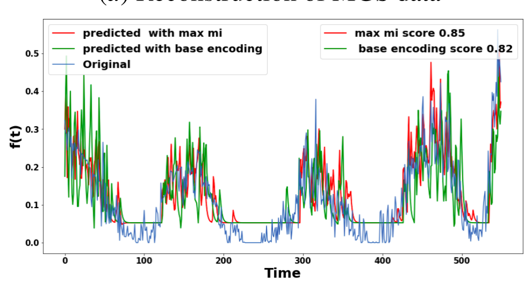
Spot data set. For the sake of brevity, we have omitted this graph for the remaining two datasets. However, we show the reconstruction performance of the network for all four datasets in Fig. 3 and report the respective scores Table 1. In each case, MI based encoding technique **achieved higher reconstruction scores (maximum 8%, 14%, 4%, 10% for MGS, Shampoo sales, Sunspot & Bike sharing dataset respectively)** than the base encoding technique - holding our hypothesis as valid. Our approach worked better than base encoding even for time series with limited observations (such as the Shampoo Sales data). This is encouraging as it points to the possibility of this approach being universally applicable for different types of time series with varying number of observations. In case of Sunspot and Bike sharing dataset (refer Fig. 3(*c*) & 3(*d*)), the network could not reconstruct low range values ($\sim 10^{-3}$) properly as these values do not create enough spikes in reservoir compared to higher range values.



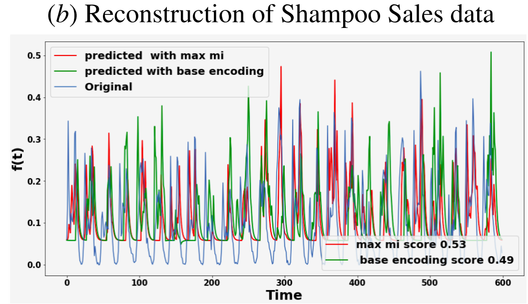
(*a*) Reconstruction of MGS data



(*b*) Reconstruction of Shampoo Sales data



(*c*) Reconstruction of Sun Spots data



(*d*) Reconstruction of Bike Sharing data

Figure 3: Experimental Results of Reconstruction

References

[1] DF Andrews and AM Herzberg. Monthly mean sunspot numbers. In *Data*, pages 67–74. Springer, 1985.

[2] Dighanchal Banerjee, Smriti Rani, Arun M. George, Arijit Chowdhury, Sounak Dey, Arijit Mukherjee, and Tapas Chakravarty. Application of spiking neural networks for action recognition from radar data. *International Joint Conference on Neural Networks*, 2020.

[3] Alexander Borst and Frédéric E Theunissen. Information theory and neural coding. *Nature neuroscience*, 2(11):947–957, 1999.

[4] Andrea Calimera, Enrico Macii, and Massimo Poncino. The human brain project and neuromorphic computing. *Functional neurology*, 28(3):191, 2013.

[5] Bruno Cessac, Horacio Rostro, Juan-Carlos Vasquez, and Thierry Viéville. How gibbs distributions may naturally arise from synaptic adaptation mechanisms. a model-based argumentation. *Journal of Statistical Physics*, 136(3):565–602, 2009.

[6] Bruno Cessac, Hélène Paugam-Moisy, and Thierry Viéville. Overview of facts and issues about neural coding by spikes. *Journal of Physiology-Paris*, 104(1-2):5–18, 2010.

[7] Hugo Gabriel Eyherabide. Neural stochastic codes, encoding and decoding. *arXiv preprint arXiv:1611.05080*, 2016.

[8] Hadi Fanaee-T and Joao Gama. Event labeling combining ensemble detectors and background knowledge. *Progress in Artificial Intelligence*, pages 1–15, 2013. ISSN 2192-6352. doi: 10.1007/s13748-013-0040-3. URL [WebLink].

[9] Arun M. George, Dighanchal Banerjee, Sounak Dey, and Arijit Mukherjee. A reservoir-based convolutional spiking neural network for gesture recognition from dvs input. *International Joint Conference on Neural Networks*, 2020.

[10] Wulfram Gerstner and Werner M Kistler. Mathematical formulations of hebbian learning. *Biological cybernetics*, 87(5-6):404–415, 2002.

[11] Hananel Hazan, Daniel J Saunders, Hassaan Khan, Darpan T Sanghavi, Hava T Siegelmann, and Robert Kozma. Bindsnet: A machine learning-oriented spiking neural networks library in python. *Frontiers in neuroinformatics*, 12:89, 2018.

[12] Alan L Hodgkin and Andrew F Huxley. A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of physiology*, 117(4):500–544, 1952.

[13] Michael Hough, Hugo De Garis, Michael Korkin, Felix Gers, and Norberto Eiji Nawa. Spiker: Analog waveform to digital spiketrain conversion in atr's artificial brain (cam-brain) project. In *International Conference on Robotics and Artificial Life*, volume 92. Citeseer, 1999.

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

[15] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004.

[16] Nikola Kasabov, Nathan Matthew Scott, Enmei Tu, Stefan Marks, Neelava Sengupta, Elisa Capecci, Muhaini Othman, Maryam Gholami Doborjeh, Norhanifah Murli, Reggio Hartono, et al. Evolving spatio-temporal data machines based on the neucube neuromorphic framework: design methodology and selected applications. *Neural Networks*, 78:1–14, 2016.

[17] L Lapicque. Recherches quantitatives sur l'excitation electrique des nerfs traitée comme une polarization. *J. Physiol. Pathol. Gen.*, 9:620–635, 1907.

[18] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Computation*, 14(11): 2531–2560, 2002.

[19] Michael C Mackey and Leon Glass. Oscillation and chaos in physiological control systems. *Science*, 197(4300):287–289, 1977.

[20] Donald H Perkel, George L Gerstein, and George P Moore. Neuronal spike trains and stochastic point processes: I. the single spike train. *Biophysical journal*, 7(4):391–418, 1967.

[21] Balint Petro, Nikola Kasabov, and Rita M Kiss. Selection and optimization of temporal spike encoding methods for spiking neural networks. *IEEE transactions on neural networks and learning systems*, 31(2): 358–370, 2019.

[22] Benjamin Schrauwen and Jan Van Campenhout. Bsa, a fast and accurate spike train encoding scheme. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 4, pages 2825–2830. IEEE, 2003.

[23] Sabyasachi Shivkumar, Richard Lange, Ankani Chattoraj, and Ralf Haefner. A probabilistic population code based on neural samples. In *Advances in Neural Information Processing Systems*, pages 7070–7079, 2018.

[24] Aboozar Taherkhani, Georgina Cosma, and TM McGinnity. Optimization of output spike train encoding for a spiking neuron based on its spatiotemporal input pattern. *IEEE Transactions on Cognitive and Developmental Systems*, 2019.

[25] Steven Wheelwright, Spyros Makridakis, and Rob J Hyndman. *Forecasting: methods and applications*. John Wiley & Sons, 1998.

[26] Anguo Zhang, Wei Zhu, and Ming Liu. Self-organizing reservoir computing based on spiking-timing dependent plasticity and intrinsic plasticity mechanisms. In *2017 Chinese Automation Congress (CAC)*, pages 6189–6193. IEEE, 2017.

Appendix A. Spike Encoders - prior art

The table below provides prior art details related to our work.

Encoder with decoder:	
Temporal-Contract (TC) [16]	Predefined (manual or auto-generated) threshold determines the occurrence of spike with the signal contrasts or changes
Hough Spiker Algorithm (HSA) [13], Bens Spiker Algorithm (BSA) [22]	1. Are event driven algorithms - a spike event is determined by de-convoluting the observed signal with a user-defined filter 2. In HSA, the convolution function produces a biased converted signal and the error quantity is minimized for the decoding part of the algorithm. 3. In BSA, a finite reconstruction filter (FIR) is used to rebuild a smooth analog signal from a digital input during decoding.
Encoder without decoder:	
Bruno et al. [6]	Proposed few deterministic metrics such as: (i) Bin metrics (e.g. by grouping spikes into bins), (ii) Convolution metrics (e.g. raster-plot metric), (iii) spike time metrics (e.g. alignment distances)
Alexander et al. [3]	1. Proved the importance of each spike to the neural code. 2. Does not support encoding of temporally dynamic stimuli.
Perkel et al. [20]	1. Proposed order dependent (e.g. serial correlation coefficient of interval lengths) and order independent (e.g. survivor function, hazard function) statistical measures. 2. Histogram of inter-spike interval was proposed as an estimator of the "actual" probability density function.
Gustaf et al. [7]	1. Generalized the idea of information theory based encoding-decoding approach. 2. Ideated use of stochastic codes to bring in variability in the neural response against a stimulus value.
Shivkumar et al. [23]	1. Described neural responses as samples from the probabilistic model previously learnt for given inputs, 2. Also used parameters of the succeeding distribution of the data.
Petro et al. [21]	1. Worked on error optimization between the original and decoded signal 2. Limited to temporal coding algorithms (having decoder).
Taherkhani et al. [24]	1. Worked towards optimizing output spike train. 2. Used supervised weight adjustment technique.

Table 2: Spike encoders and decoders and their features

Appendix B. Poisson Encoding (Base encoding)

As mentioned Section 1, the real-valued stimulus data are usually encoded into spike trains using non-temporal neural coding schemes (e.g. Poisson, Bernoulli etc.). In our experiment, we have used Poisson encoding to create the initial spike train. For a Poisson process, the probability of observing exactly n spikes in a time interval (t_1,t_2) is given by:

$$P(n\ spikes\ during\ (t_1, t_2)) = e^{(<n>)} \frac{| < n > |^n}{n!} \tag{3}$$

where $< n >$ is the average spike count given by

$$< n >= \int_{t_1}^{t_2} r(t)dt \tag{4}$$

r(t) being the instantaneous firing rate. Assuming that r(t) varies very slowly in the small time sub-interval δt, r(t) is sampled to produce discrete rate value r[i]. We know that any Poisson process can be approximated by infinitely many Bernoulli trials and a Bernoulli trial can be simulated by taking a uniform draw x[i] at each time step i, thereby creating a spike T[i] = 1 if x[i] < $r[i]\delta t$ and T[i] = 0 if x[i] $\geq r[i]\delta t$.

Appendix C. SNN, Reservoir & Information theory- A Brief Background

Spiking Neural Networks, often considered as the 3rd generation of neural networks, work with a philosophy different from traditional artificial neural networks (ANN). The neurons in SNN operate based on the mathematical models of actual biological spiking neurons which undergo chemical changes (flow of ionic components) on receiving any external stimuli. Depending upon the quantum of this chemical change, the electrical potential changes resulting in one or more spikes propagating through the neurons, thereby carrying information for further processing. Different mathematical models describing this spiking phenomena have been proposed, such as the Hodgkin-Huxley model [12] (most detailed and complex model), Izikevich model [14], Leaky Integrate and Fire (LIF) model [17] (simple and mostly used) etc. In LIF (the model we use in our work) the membrane potential V of a neuron cell, at any point in time can be mathematically described by Equation 5:

$$\tau_{mem} \frac{dV}{dt} = (V_{rest} - V) + g_e(E_{exc} - V) + g_i(E_{inh} - V) \tag{5}$$

The resting potential, V_{rest} is a point attractor towards which the membrane potential tends to evolve. Without any input (aka stimuli) from other pre-synaptic neurons, membrane potential remains at V_{rest}. Synapses between two consecutive neurons can be excitatory and inhibitory in nature; E_{exc} and E_{inh} represents equilibrium potentials for those synapses respectively. Functionality of synapses are represented via conductance values namely g_e, the excitatory conductance and g_i, the inhibitory conductance. When a pre-synaptic neuron spikes, the conductance of the synapse increases in magnitude. Excitatory pre-synaptic neurons increase the membrane potential whereas inhibitory pre-synaptic neurons decrease it. When the membrane potential crosses a threshold, V_{thresh}, a spike is generated by that neuron. Communication between neurons through synapses occurs via these asynchronous time events called spikes. The dynamics of excitatory and inhibitory conductance is modelled as per Equation 6.

$$\tau_e \frac{dg_e}{dt} = -g_e \quad \text{and,} \quad \tau_i \frac{dg_i}{dt} = -g_i \tag{6}$$

The pre-synaptic trace x_{pre} for each synapse keeps track of the activity of the pre-synaptic neuron and the post-synaptic trace x_{post} tracks the activity of the post-synaptic neuron. Each trace decays exponentially with time as shown in the Equation 7 with synaptic trace decay constants τ_{pre} and τ_{post}. When a spike occurs at a pre or post-synaptic neuron, it's trace is increased in magnitude by a constant value 1.

$$\tau_{pre}\frac{dx_{pre}}{dt} = -x_{pre} \quad \text{and,} \quad \tau_{post}\frac{dx_{post}}{dt} = -x_{post} \tag{7}$$

In the computing domain, for an SNN model to process real world values, the input must be encoded as spike trains made up of 0s and 1s, and typically two types of encoding are used: *rate coding* and *temporal coding*. It has been found that computational tasks like gesture recognition [9], time series prediction [26] etc. i.e. tasks involving temporally varying signals such as video/audio, recurrently connected spiking neurons, popularly called *reservoir*, work efficiently when used along with these encoding methods.

This concept of *reservoir computing* was independently researched by two research groups resulting in two different models - the Echo State Network (ESN) [15] and the Liquid State Machine (LSM) [18]. ESNs focused on rate based neurons and LSMs on spiking neurons - the latter being more relevant for our work. Using LSM, the temporal features present in the input are extracted by a *reservoir*, called as *liquid* and output of this liquid is trained to produce the desired activity based on a learning rule. A reservoir is a population of random and recurrently connected spiking neurons with N_{exc} and N_{inh} number of excitatory and inhibitory connections respectively. Amongst those neurons there are a total of N_{rec} number of recurrent connections. These group of neurons are able to learn patterns from the temporal dynamics of data and to recreate it.

During the course of our work, We found that the performance of reservoir (and in general, any spiking network) is highly dependent on efficiency of encoding of real values, i.e. on the amount of information preserved during spike encoding. To our knowledge, there is no standard quantitative method that can ensure maximum information content in the encoded spike train. We thus introduced the idea of maximizing the mutual information between these two entities so that information preservation is ensured.

C.1. Information theoretic background

To this end, we used the concept of *Information Theory*, a probability based mathematical framework that was first used for quantifying information transmission in communication systems around 1940s. It can also be used to quantify how much information about any external stimulus is carried in a neural response. One way to quantify this information is through entropy of a random variable ($H(X)$) which can be calculated as per Equation 8, where $P_X(x)$ is the probability of the random variable X taking the value x. The entropy can be thought of as a measure of the disorder inherent in the variable X, given the size of the number of different values X can take.

$$H(X) = -\sum_{x\in\mathcal{X}} P_X(x)\log P_X(x) \tag{8}$$

The mutual information (M.I.) is a measure to gain insight about the information that one piece of data (i.e. one random variable) carries about the other. Mutual information is mathematically expressed as:

$$I(X;Y) = \sum_{(x,y)\in\mathcal{X}\times\mathcal{Y}} P_{XY}(x,y)\log\frac{P_{XY}(x,y)}{P_X(x)P_Y(y)} \tag{9}$$

In fact, it gives us insight about the answer to the question "how far are X and Y from being independent of each other?". Through some simple algebra it can be shown that:

$$I(X;Y) = H(X) - H(X|Y). \tag{10}$$

While encoding an external stimulus signal into spike trains, MI can be used to measure how much information the spike train carries about the original data.

Appendix D. Parameter values of LIF neuron model and Reservoir

Different parameter values for the LIF model of spiking neurons (such as V_{thresh}, V_{rest} etc.) and of reservoir network (such as N_{exc}, N_{inh}) which have been used in our experiment are mentioned in Table 3 below.

Parameter	Value	Parameter	Value	Parameter	Value
V_{thresh}	-52.0 mV	τ_{mem}	100 ms	N_{exc}	1000
V_{rest}	-65.0 mV	τ_{pre}	100 ms	N_{inh}	250
		τ_{post}	100 ms	N_{rec}	500

Table 3: Parameter values for spiking neuron reservoir

Appendix E. Evaluation Metric

Coefficient of Determination or R^2 Score is a metric used to measure the goodness of statistical regression models. It quantifies the amount of variance in the error of predicted value that remains unexplained with respect to the the amount of variance in the input data. In our case, an R^2 Score of 0.xx signifies that xx% of the variability of the reconstructed signal is accounted for (with respect to original signal) by the model and that (100 - xx)% of the variability is still unaccounted for. R^2 Score ranges from (-infinity, 1.0] with 1.0 represents perfect correlation between reconstructed and the original signal.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{11}$$

where SS_{res} is the residual sum of squares and and SS_{tot} is the total sum of squares representing the variance of the original signal, defined as Eqn 12.

$$SS_{res} = -\sum_i (pred_i - true_i)^2 \quad \text{and,} \quad SS_{tot} = -\sum_i (true_i - mean(true))^2 \tag{12}$$