Spike Encoding for Pattern Recognition: Comparing Cerebellum Granular Layer Encoding and BSA algorithms

Chaitanya Medini
Amrita School of Biotechnology
Amrita Vishwa Vidyapeetham (Amrita University)
Clappana (P.O.), Kollam, Kerala
krishnachaitanya@am.amrita.edu

Asha Vijayan
Amrita School of Biotechnology
Amrita Vishwa Vidyapeetham (Amrita University)
Clappana (P.O.), Kollam, Kerala
ashavijayan@am.amrita.edu

Ritu Maria Zacharia
Amrita School of Engineering
Amrita Vishwa Vidyapeetham (Amrita University)
Clappana (P.O.), Kollam, Kerala
ritu.mariaz@gmail.com

Lekshmi Priya Rajagopal
Amrita School of Engineering
Amrita Vishwa Vidyapeetham (Amrita University)
Clappana (P.O.), Kollam, Kerala
lekshmipriya310@gmail.com

Bipin Nair
Amrita School of Biotechnology
Amrita Vishwa Vidyapeetham (Amrita University)
Clappana (P.O.), Kollam, Kerala
bipin@amrita.edu

Shyam Diwakar, *SM*IEEE
Amrita School of Biotechnology
Amrita Vishwa Vidyapeetham (Amrita University),
Clappana (P.O.), Kollam, Kerala
shyam@amrita.edu

***Abstract*—Spiking neural encoding models allow classification of real world tasks to suit for brain-machine interfaces in addition to serving as internal models. We developed a new spike encoding model inspired from cerebellum granular layer and tested different classification techniques like SVM, Naïve Bayes, MLP for training spiking neural networks to perform pattern recognition tasks on encoded datasets. As a precursor to spiking network-based pattern recognition, in this study, real world datasets were encoded into spike trains. The objective of this study was to encode information from datasets into spiking neuron patterns that were relevant for spiking neural networks and for conventional machine learning algorithms. In this initial study, we present a new approach similar to cerebellum granular layer encoding and compared it with BSA encoding techniques. We have also compared the efficiency of the encoded dataset with different datasets and with standard machine learning algorithms.**

Keywords—nature-inspired computing; classification; machine learning; encoding; spiking neuron

I. Introduction

Cerebellum has always been considered as a look-up table [1] and as a pattern classifier [2] coordinating controlled limb movement [3] and learning [4]. For this coordinated learning process, the cerebellum acts an error-correction mechanism where it detects novelty between intended and actual movement patterns [2] [5]. Based on the modular organization and functioning of cerebellum, there are several models like spiking neural networks (SNN) [6], [7], artificial neural networks [8], [9] and internal models [10], [11] which have been used in control theory and artificial intelligence. Robotics is another key area where these models had been applied to perform real-time classification of bio-signals like electromyographic (EMG) and electroencephalographic (EEG) signals [12]. These signals were used as non-invasive measures for prosthetic limb movement which could restore the lost motor functions [13]. For this application, SNNs are usually employed, where the input data is first encoded [14] in the form of spikes or spike trains. There are different neural encoding schemes [14]–[18] to convert real time data into spike trains. These spike trains are the temporal sequence of action potentials generated by a neuron which can be modeled using different neuronal models like leaky integrate and fire (LIF) model and adaptive exponential integrate and fire (AdEx) [19] model.

Previous studies on pattern classifications showed that cerebellar micro-circuitry was involved in learning new sensori-motor associations through adaptation [2], [4]. This was followed by the introduction of perceptron like model with different learning algorithms [20]–[22] to classify the incoming spike generated using LIF model. Recent studies have shown that classification of EEG based signals for different disease conditions like epilepsy can be obtained with the help of different encoding schemes as well as supervised learning algorithms [17], [23], [24]. Learning algorithms like Finite Precision (FP) learning and High-Threshold Projection (HTP)

978-1-4799-8792-4/15/$31.00 ©2015 IEEE

have been used to generate output spikes with specified temporal tolerance for associated inputs [25].

The goal of this paper was to model spiking neuron patterns for datasets to be classified by spiking neural networks and conventional machine learning algorithms. Toward this goal, a new encoding scheme was developed that could reconstruct spiking patterns as seen in cerebellum granular layer. The spike trains or spikes obtained from our encoding algorithm is given as the input to the SNN and the output spikes obtained from the SNN is decoded back to real-world data.

In this paper, we address how a bio-inspired representation was developed to encode data as patterns of action potentials or spikes. In order to compare, a preexisting encoding-decoding scheme was implemented based on Bens Spike Algorithm (BSA) [17]. Real world datasets were converted into patterns similar to *in vitro* and *in vivo* rat cerebellum granule cell firing patterns [26]. Eighteen different firing patterns were modeled using an Adaptive Exponential leaky integrate and fire (AdEx) model [19]. These spike patterns were given as input to a feed-forward spiking neural network for pattern recognition. These encoding schemes were tested on some machine learning datasets obtained from UCI Machine Learning repository [27].

II. METHODS

A. Encoding of real world data into neural information

For encoding datasets to spikes or spike trains, different encoding schemes [14], [17], [18] including spiking neural network and impulse response filters models were used.

1) BSA Encoding Scheme

Bens Spike Algorithm uses rate encoding employing average number of spikes per unit time. The algorithm included a Finite Impulse Response (FIR) reconstruction filter and a threshold value. A fixed threshold value of 0.86 has been used for implementing BSA [28]. BSA encoding algorithm works by converting real world data or signals into spike trains. Real world data has been normalized using the min-max normalization (1) method [29] to the range of filter values.

$$\text{Norm}_{\text{data}}=\frac{\text{data- min(data)}}{\text{max(data)- min(data)}}*(\text{new\_max-new\_min})+\text{new\_min} \quad (1)$$

Here $\text{Norm}_{\text{data}}$ refers to the normalized value, data refers to the current value, min(data) and max(data) refers to the minimum and maximum value, new_max and new_min refers to the maximum and minimum values to which the current values are normalized to.

The FIR filter minimizes the mean square error between the input and estimated stimulus. Estimated stimulus (s_{est}) was calculated using (2), as shown in [17].

$$s_{est} = (h * x)(t) = \int_{-\infty}^{\infty} x(t-\tau)h(\tau)d\tau = \sum_{k=1}^{N} h(t-t_k) \quad (2)$$

Here h(t) refers to filter response, t refers to the current time, t_k refers to the firing times of the neuron, τ is time step, N refers to the number of filter taps and x(t) was found by (3)

$$x(t) = \sum_{k=1}^{N} \delta(t-t_k) \quad (3)$$

After estimating the stimulus, two error metrics (4), (5) were calculated at every time point τ, for smoothening the frequency and amplitude characteristics in BSA [28].

$$Error1 = \sum_{k=0}^{N} abs(s(k+\tau) - h(k)) \quad (4)$$

$$Error2 = \sum_{k=0}^{N} abs(s(k+\tau)) \quad (5)$$

Where s denotes the input stimulus.

The input data was subtracted repeatedly using FIR filters with a 1 bit shift per iteration until all the results became zero. If the first error metric (i.e. error1) was less than the second one (i.e. error2) minus threshold, then a spike was emitted and subtracted the FIR filter from the input dataset. Otherwise, no spikes were emitted. The standard filter values used were <1, 4, 9, 5, -2> as indicated in [24]. Thus, the spiking patterns were obtained in the binary form.

2) BSA using different filters

By using FDA (Filter Design and Analysis) tool in Matlab [30], different FIR filter values were generated. Equiripple [30] as well as Window FIR [30] design methods with different response types such as low-pass, high-pass and band-pass were used to generate the filter values of the order 8. Filter coefficients were generated and normalized to a range 1-10, by using min-max normalization method [29]. These filter values were used to generate spike patterns with the BSA encoding.

3) Look-up table based encoding

As a novelty, cerebellum-inspired encoding scheme was used which mapped inputs to known physiological conditions. *In vivo* and *in vitro* granule neuron firing patterns obtained from [26] have been used for the spiking behavior. Adaptive exponential integrate and fire (AdEx) spiking neuron model has been used to mimic these neuronal firing behaviors [26], [31]. This model contains only two differential equations (6) and (7) with a reset condition (8) as shown in [19]. Equation (6) refers to membrane potential V being computed, while (7) refers to adaptation current w being calculated when a current I was injected.

$$C(\frac{dV}{dt}) = -g_L(V-E_L) + g_L\Delta T \exp(\frac{V-V_T}{\Delta T}) + I - w \quad (6)$$

$$\tau_w(\frac{dw}{dt}) = a(V-E_l) - w \quad (7)$$

Here C refers to total capacitance, g_L refers to total leak conductance, E_L refers to effective rest potential, ΔT refers to threshold slope factor, V_T refers to effective threshold potential, V_r refers to reset potential, τ_w refers to adaptation time constant, a refers to adaptation coupling parameter, b refers to spike triggered adaptation.

When the membrane potential crossed the threshold, a reset function was applied and the membrane potential as well as adaptation current variables were reset with the following conditions:

$$if\ V > 0, then \begin{cases} V \rightarrow V_r \\ w = w + b \end{cases} \quad (8)$$

Depending on different parameter values, different firing patterns were generated. The focus was only on the granular layer network [26], so tonic spiking patterns were generated. The parameter values to reproduce the tonic spiking pattern has been shown in Table 1. The parameters I and E_l changed depending on each input

B. *Simple Spiking Neural Network*

Simple spiking neural model helps to understand the neural dynamics and the input-output associations. In several studies, different learning rules like Multi-SpikeProp [32] and RProp [33] were used with SNNs. The neural network was created using simple AdEx [19] model which contain 2 layers, the first layer resembling the granular layer and the second layer resembling a single Purkinje neuron, whose output was computed for classification. A similar study [25] with LIF neuron was done to find the number of associations between the input-output spike trains using HTP algorithm and FP learning algorithm. The output from the encoding scheme (spike trains) were distributed to the feed-forward part of SNN where each pattern corresponded to a granule cell firing pattern.

TABLE I. PARAMETER VALUES

C	g_L	V_T	ΔT	a	b	V_r	τ_w
200	10	-50	2	2	0	-58	30

As in rat cerebellum, each input afferent resulted in post synaptic potential (PSP) [25] estimated for a total time period *T* (200ms)

$$\text{PSP} = \omega^T x_i(t) = \Sigma \omega_i x_i(t) \tag{9}$$

Where, each input afferent, *i*, emitted spikes at time $\{t_i\}$ and ω_i was afferent's synaptic efficacy and $x_i(t)$ was computed as:

$$x_i(t) = \sum_{t_i<t} u(t - t_i) \tag{10}$$

Where, $u(t) = U_0(e^{-\left(\frac{t}{\tau_m}\right)} - e^{-\left(\frac{t}{\tau_s}\right)})$, with τ_m being the membrane constant and τ_s being the synaptic time constant and given as $\tau_m = 4\tau_s$ [34]. τ_m value was chosen as 30ms for the study within a time window 1ms. The generated PSP was used to compute Purkinje cell response. After spiking, the membrane equation was reset and the supra-threshold potential (U(t)) was estimated as in (11).

$$U(t) = \omega^T x(t) - U_{thr} x_{reset}(t) \tag{11}$$

Here U_{thr} refers to threshold potential. Modeling the reset to zero, x_{reset} was computed by,

$$x_{reset}(t) = \sum_{t_{spike}<t} u_r(t - t_{spike}) \tag{12}$$

Here, $\{t_{spike}\}$ refers to the output spike times and u_r(t) was found as reported in (13).

$$u_r(t) = e^{-\left(\frac{t}{\tau_m}\right)} \tag{13}$$

An adapted version of the High Threshold Projection (HTP) algorithm [25] was used as a learning rule to update weights. The algorithmic implementation of HTP in our encoding includes 3 conditions in order to have an output spike:

(1) $U(t_d) = U_{thr}$, where U_{thr} was the threshold potential (linear equality constraint)

(2) $\frac{dU}{dt(td)}$>0 for all t_d (linear inequality constraint)

(3) $U(t) < U_{thr}$for all t except t_d (linear inequality constraint)

Feed forward Spiking Algorithm (adaptation of HTP Algorithm [25]):

```
Begin
        1.  Initialize the weight $\omega$ with random numbers
        2.  For every pattern in the training set
            a. Present the pattern to the network and calculate
               the PSP using (9)
            b. For output neuron
                 i. Create a set of error times, $\{t_{err}\}$ and find
                    the error
                ii. Stop if no error found i.e. all equality and
                    inequality constraints are satisfied.
               iii. Construct a set of input labeled patterns,
                    {(x,y)}, where x=x($t_{err}$), y=-1 if $t_{err}$ is
                    above the suprathreshold, else x=
                    diff(x($t_{err}$), y=+1 if $t_{err}$=$t_d$.
                iv. Add the constructed set to x.
            c. End
3.   End
```

C. *Decoding*

Here a BSA decoding scheme [28] using the same FIR filters was used to recode output data. The input data was recovered from the spiking patterns, which was estimated in the binary form. A mask filter with values 0 and length same as that of FIR filter was used. The BSA decoding algorithm was computed from the spike train: first, if the value was 1, FIR filter was applied; else, mask filter was used. A 1-bit shift per iteration was used and filters applied for all rows until the end of spike train. The columns were summed to get the output.

III. RESULTS

A. *Encoding of signals to spike train*

We have developed a cerebellum inspired model to understand the electrophysiological properties of neuron in terms of what it does as a function or simply the neural excitability. For this, different encoding schemes were used.

Encodings were tested on different datasets like weather dataset [27], lung cancer dataset [27] and a robotic arm dataset [35] to test encoding.

1) Encoding using BSA

Spiking patterns were generated as 1's and 0's using BSA encoding. Here 1's represent action potentials and 0's represent non spiking information. Time period used to generate a single spike was 100ms [36]. Fig. 1A shows the firing patterns generated by BSA encoding scheme.

Using the BSA encoding scheme, the dataset was normalized within the range of filter values since classifiers and clustering mechanisms look at the generalization capability for non-specific dataset and by using spike encodings, the dataset was made non-specific. A threshold value of 0.86 [28] was used. By using FDA (Filter Design and Analysis) tool, we designed different FIR filter values by using Equiripple [37] as well as Window [37] FIR design methods and also different response types such as low-pass, high-pass and band-pass. Generated spike patterns from the designed filters are in binary data format. By using AdEx spiking neuron model, value 1 was mapped to single spike and the value 0 to no spikes. Spike patterns were generated within 100ms.

2) *Granule layer encoding using biophysical model*

In this method, 18 different granule cell firing patterns from biophysical model were generated. Based on each input different patterns are generated within 200ms. (Fig. 1B)

In this encoding scheme, dataset normalization was done within the range of 1 and 18 since 18 different *in vivo* and *in vitro* granule cell firing patterns that can be seen in physiological conditions were used. Unlike the BSA, which used only 1 parameter to calculate the filter value, here we used a lookup kind of approach which changes the nine AdEx parameters for every pattern. Normalized input was mapped to different physiological conditions as seen in neurophysiology experiments and generated different spiking patterns using AdEx model. Fig.1 showed the two different encoding schemes used to generate spike patterns. Fig. 1A shows the spikes generated with <1, 4, 9, 5, -2> filter using BSA encoding scheme and Fig. 1B shows the spikes generated using granule layer encoding scheme for the inputs {2,1,1,2} and {3,2,1,2} in a weather dataset.

Different algorithms were tested on the dataset with 66% split to compare the efficiency of the datasets used after the encoding. Classifiers on lung cancer dataset demonstrated higher efficiency compared to robotic arm dataset and weather dataset.

Algorithms like linear SVM and nonlinear SVM with polynomial kernels of order 3 and 4, Naïve Bayes, J48, Random forest and MLP with different hidden layers and learning rates on the real dataset and encoded dataset with different filters after normalization. Filter1 = [1, 4, 9, 5, -2] and filter2 = [2.8037, 1.0000, 4.4688, 7.9375, 10.0000] were used. For the lung cancer dataset, non-linear SVM and MLP (with 11 hidden layers and 0.023 learning rate) showed higher accuracy (Fig. 2) while for the robotic arm dataset, random forest and MLP (with 3 hidden layers and 0.1 learning rate) showed higher accuracy. Weather dataset showed a consistent rate of 60% accuracy for all the classifiers except random forest.

Classification results are shown in Table II. Fig. 2 shows the comparison of different machine learning algorithms on lung cancer dataset.

B. *Spiking Perceptron model for encoded dataset*

1) *Feed-forward Dynamics*

We have modelled the adaptation based on Hebbian-like learning process to mimic the dynamics observed in the molecular layer of cerebellum.

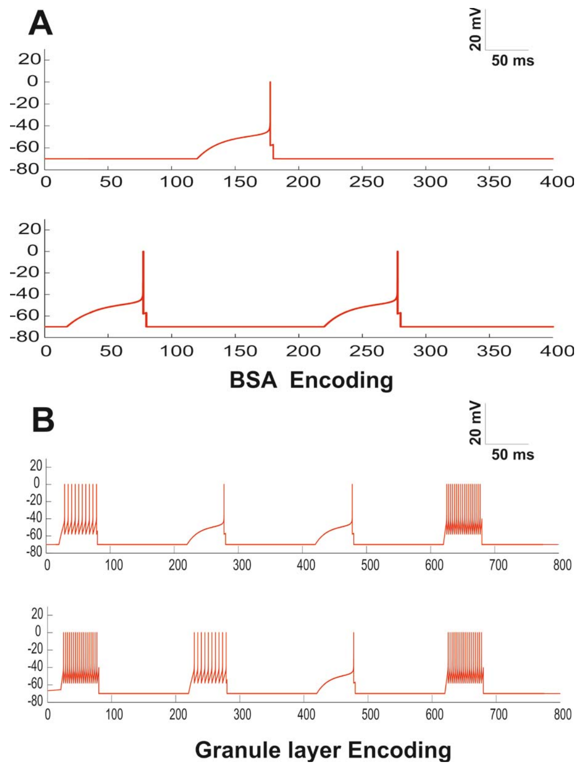

Fig. 1. Encoding Using BSA and Granule Layer Encoding. Generation of spike trains as potential (y-axis) with time (x-axis). A shows encoded spike patterns using BSA encoding for two different input patterns. B shows encoded spike patterns using granule layer encoding for the same two input patterns.

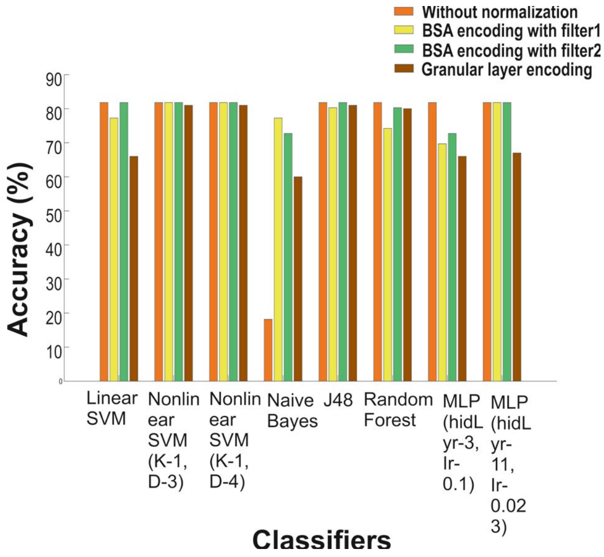

Fig. 2: Comparison of training efficiency on lung cancer dataset for different classifiers. SVM and MLP (11 hidden layers and 0.023 learning rate), showed higher efficiency compared to other machine learning methods.

For this, a feed-forward spiking neural network with 2 layers were built, the first one being the input layer, which resembles the granule layer and the second resembling the Purkinje layer which received the input from the encoding algorithm in the form of spike patterns as seen in the cerebellar micro-circuitry. The incoming spike activities were summed up to reach a threshold (U_thr = 40mV), to generate a spike. This was based on a simple perceptron model where each neuron performed a weighted sum of its incoming spikes and generated output [9].

2) *Reproducing Purkinje Neuron like Firing Behavior*

Spike trains from the encoding scheme were applied to the spiking neuron network. A perceptron-like activity based network was reconstructed as seen in the Purkinje Neuron model (Fig 4) with spikes as 1 and rest as 0s. Spike patterns were applied to the feed-forward dynamics (9), where the output pattern was generated from these input afferents based on the Perceptron. The Purkinje-like firing pattern was simulated which was taken as the output of our system. That is, after applying the feed-forward dynamics, the output should be either as seen in Fig 4.

IV. CONCLUSION

We have reconstructed an encoding process via spiking behavior as seen in the cerebellar circuits. In this effort, different encoding schemes like granule layer encoding scheme and BSA encoding schemes were used to encode datasets as spike trains. In BSA encoding scheme, different filters were used including those created with the FDA tool. Matching physiological patterns to represent granule layer encoding, helped convert machine learning datasets into spike trains, which could be used by prediction-correction scheme based brain-computer interfaces (BCIs).

TABLE II. Classification Accuracy of Different Conventional Machine Learning Approaches

Algorithm	**Classification Accuracy (%)**											
	Robotic Arm Dataset				**Lung-Cancer Dataset**				**Weather Dataset**			
	Without normalization	***BSA Encoding with normalization***		***Granule layer Encoding***	***Without normalization***	***BSA Encoding with normalization***		***Granule layer Encoding***	***Without normalization***	***Encoding with normalization***		***Granule layer Encoding***
		Filter1	***Filter2***			***Filter1***	***Filter2***			***Filter1***	***Filter2***	
SVM (linear)	60.53	55.26	57.89	73	81.82	77.27	81.82	67	60	60	40	60
SVM (nonlinear kernel-3)	50	50	50	50	81.82	81.82	81.82	81	60	60	60	60
SVM (nonlinear kernel-4)	50	50	50	50	81.82	81.82	81.82	81	60	60	60	60
Naïve Bayes	68.42	55.26	52.63	68	18.18	77.27	72.73	60	60	60	40	60
J48	50	60.53	52.63	55	81.82	80.30	81.82	81	60	60	40	40
Random Forest	65.79	63.16	55.26	71	81.82	74.24	80.30	80	60	40	40	80
MLP (Hidden Layers-3, Learning Rate-0.1, Training Time- 100)	60.53	63.16	52.63	71	81.82	69.70	72.73	66	60	60	60	60
MLP (Hidden Layers-11, Learning Rate-0.023, Training Time- 100)	60.53	50	55.26	71	81.82	81.82	81.82	67	60	60	60	60

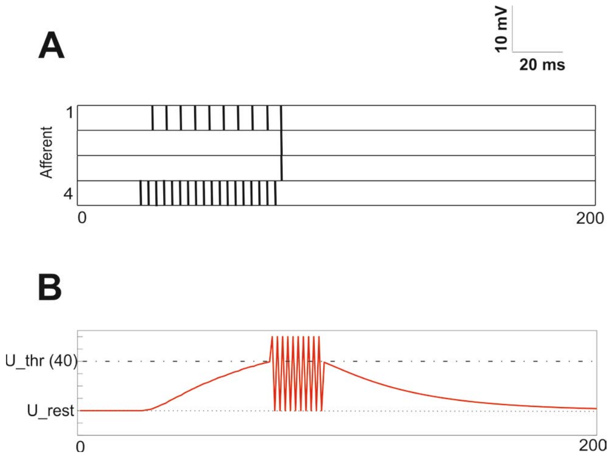

Fig. 3. Feed-Forward Dynamics. A shows an example of input pattern where each pattern showed the activity of each input afferent, where the vertical bar represented the spike timing. B shows the voltage trace after inputs were summed up which they received from the input afferents according to the feed-forward dynamics with repect to PSP (y-axis) vs time (x-axis). When it reached above a threshold (U_thr), output spikes were being generated.

Through the encoding schemes, we have tried to abstract the sparse encoding as seen in cerebellum granular layer using bursting and spiking patterns. The role of inter-spike intervals are yet to be explored. While comparing encoded patterns and datasets, lung cancer dataset showed better efficacy than on other datasets suggesting the more the features a unique encoded pattern is formed which helps in classification. Linear and nonlinear SVM and MLP showed better accuracy than other classifiers. The comparison of the proposed model with the exsisting model in terms of computation is still to be explored.

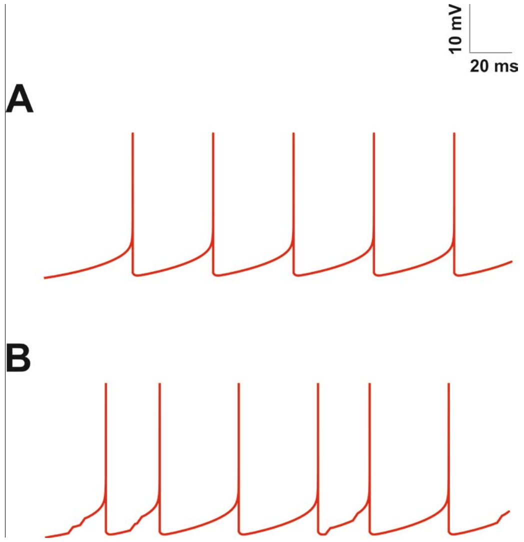

Fig. 4. Perceptron like Firing Activity when membrane potential (y-axis) varied with respect to time (x-axis). Using encoded spiking information from datasets, Purkinje like firing behavior was simulated using perceptron concept. A shows Purkinje cell like firing pattern when there was no input being received. B shows the event when there is spike or action potential which was taken as 1.

A feed-forward spiking neural network was reconstructed along with this as a goal to reconstruct the cerebellum like network where the input is given in the form of spikes as seen in the cerebellar micro-circuitry where the input propagate through each layer and an output is being produced. We are looking at the generalization capabilities of spiking classifiers for non-specific data using spike encoding. Currently an extension of the work would be to implement the HTP algorithm [25] as a learning rule which updates a set of weights which could be used for input-output associations for the feed forward spiking neural network. The results have implications that could be used for several direct and non-invasive BCI requiring physiological signals to be used for operating devices [38].

Acknowledgment

This work derives direction and ideas from the Chancellor of Amrita University, Sri Mata Amritanandamayi Devi. This work is supported by grants SR/CSI/60/2011, Department of Biotechnology (DBT) grant BT/PR/5142/MED/30/764/2012, Government of India.

References

[1] J. S. Albus, “A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller(CMAC),” *J. Dyn. Syst. Meas. Control*, 1975.

[2] J. S. Albus, “A Theory of Cerebellar Function,” *Math. Biosci.*, vol. 10, no. 1/2, pp. 25–61, 1971.

[3] N. L. Cerminara, R. Apps, and D. E. Marple-Horvat, “An internal model of a moving visual target in the lateral cerebellum.,” *J. Physiol.*, vol. 587, pp. 429–442, 2009.

[4] D. Marr, “A theory of cerebellar cortex.,” *J. Physiol.*, vol. 202, no. 2, pp. 437–70, Jun. 1969.

[5] D. Purves, G. J. Augustine, D. Fitizpatrick, W. C. Hall, A.-S. LaMantia, J. O. McNamara, and S. M. Williams, *Neuroscience Third Edition*, Third Edit. 2004.

[6] R. R. Carrillo, E. Ros, C. Boucheny, and O. J.-M. D. Coenen, “A real-time spiking cerebellum model for learning robot control.,” *Biosystems.*, vol. 94, no. 1–2, pp. 18–27, 2008.

[7] X. Jin, M. Luján, L. A. Plana, S. Davies, S. Temple, S. B. Furber, M. Lujan, L. A. Plana, S. Davies, S. Temple, and S. B. Furber, “Modeling Spiking Neural Networks on SpiNNaker,” in *Computing in Science & Engineering*, 2010, vol. 12, no. 5, pp. 91–97.

[8] N. Schweighofer, J. Spoelstra, M. A.Arbib, and M. Kawato, “Role of the cerebellum in reaching movements in humans. II. A neural model of the intermediate cerebellum,” *Eur. J. Neurosci.*, vol. 10, no. 1, pp. 95–105, 1998.

[9] F. Rosenblatt, *Principles of Neurodynamics*. 1962.

[10] M. Kawato, T. Kuroda, H. Imamizu, E. Nakano, S. Miyauchi, and T. Yoshioka, “Internal forward models in the cerebellum: fMRI study on grip force and load force coupling,” *Prog. Brain Res.*, vol. 142, pp. 171–188, 2003.

[11] D. Wolpert and M. Kawato, “Multiple Paried Forward and Inverse Models for Motor Control,” *Neural Networks*, vol. 11, pp. 1317–1329, 1998.

[12] S. Diwakar, S. Bodda, C. Nutakki, A. Vijayan, and K. Achuthan, “Neural Control using EEG as a BCI Technique for Low Cost Prosthetic Arms,” in *6th International Conference on Neural Computation Theory and Applications NCTA 2014*, 2014.

[13] B. Crawford, K. Miller, P. Shenoy, and R. Rao, “Real-Time Classification of Electromyographic Signals for Robotic Control,” in *The Twentieth National Conference on Artificial Intelligence and the Seventeenth Innovative Applications of Artificial Intelligence Conference(AAAI)*, 2005, pp. 523–528.

[14] T. Iakymchuk, A. Rosado-munoz, M. Bataller-mompean, J. F. Guerrero-martinez, J. V Frances, A. Rosado-mu, M. Bataller-mompe, J. F. Guerrero-mart, and J. V Franc, “Frequency spike encoding using Gabor-like receptive fields,” *19th World Congr. Int. Fed. Autom. Control.*, pp. 701–706, 2014.

[15] E. T. Rolls and A. Treves, “The neuronal encoding of information in the brain,” *Prog. Neurobiol.*, vol. 95, no. 3, pp. 448–490, 2011.

[16] S. Koyama, “On the relation between encoding and decoding of neuronal spikes,” *BMC Neurosci.*, vol. 12, no. Suppl 1, p. P177, 2011.

[17] B. Schrauwen and J. Van Campenhout, “BSA , a Fast and Accurate Spike Train Encoding Scheme,” in *International Joint Conference on Neural Networks*, 2003, no. 1, pp. 2825–2830.

[18] Y. Gao, M. J. Black, E. Bienenstock, W. Wu, and J. P. Donoghue, “A quantitative comparison of linear and non-linear models of motor cortical activity for the encoding and decoding of arm motions,” in *Proceedings of the 1st International IEEE EMBS Conference on Neural Engineering*, 2003, pp. 189–192.

[19] R. Naud, N. Marcille, C. Clopath, and W. Gerstner, “Firing patterns in the adaptive exponential integrate-and-fire model.,” *Biol. Cybern.*, vol. 99, no. 4–5, pp. 335–347, Nov. 2008.

[20] N. Brunel, V. Hakim, P. Isope, J. P. Nadal, and B. Barbour, “Optimal information storage and the distribution of synaptic weights: Perceptron versus Purkinje cell,” *Neuron*, vol. 43, pp. 745–757, 2004.

[21] C. Clopath, J. P. Nadal, and N. Brunel, “Storage of correlated patterns in standard and bistable Purkinje cell models,” *PLoS Comput. Biol.*, vol. 8, no. 4, pp. 1–10, 2012.

[22] R. Rubin, R. Monasson, and H. Sompolinsky, “Theory of spike timing based neural classifiers,” p. 4, 2010.

[23] S. Ghosh-Dastidar and H. Adeli, “A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection,” *Neural Networks*, vol. 22, no. 10, pp. 1419–1431, 2009.

[24] N. Nuntalid, K. Dhoble, and N. Kasabov, “EEG classification with BSA spike encoding algorithm and evolving probabilistic spiking neural network,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7062 LNCS, no. PART 1, pp. 451–460, 2011.

[25] R.-M. Memmesheimer, R. Rubin, B. P. Olveczky, and H. Sompolinsky, “Learning Precisely Timed Spikes,” *Neuron*, vol. 82, pp. 1–14, 2014.

[26] C. Medini, A. Vijayan, E. D’Angelo, B. Nair, and S. Diwakar, “Computationally Efficient Bio-realistic Reconstructions of

Cerebellar Neuron Spiking Patterns," in *Proceedings of the 2014 International Conference on Interdisciplinary Advances in Applied Computing - ICONIAAC '14*, 2010.

[27] M Lichman, "UCI Machine Learning Repository," *University of California, Irvine, School of Information and Computer Sciences*, 2013. [Online]. Available: http://archive.ics.uci.edu/ml.

[28] N. Nuntalid, "Evolving Probabilistic Spiking Neural Networks for Modelling and Pattern Recognition of Spatio-temporal Data on the Case Study of Electroencephalography (EEG) Brain Data Evolving Probabilistic Spiking Neural Networks for Modelling and Pattern Recognitio," 2012.

[29] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Second Edi. Morgan Kaufmann, 2006.

[30] "MATLAB AND SIGNAL PROCESSING TOOLBOX RELEASE 2012b, The MathWorks, Inc., Natick, Massacheusetts, United States." .

[31] C. Medini, B. Nair, E. D'Angelo, G. Naldi, and S. Diwakar, "Modeling spike-train processing in the cerebellum granular layer and changes in plasticity reveal single neuron effects in neural ensembles.," *Comput. Intell. Neurosci.*, vol. 2012, p. 359529, Jan. 2012.

[32] S. Ghosh-Dastidar and H. Adeli, "Improved Spiking Neural Networks for EEG Classification and Epilepsy and Seizure Detection," *Integr. Comput. Aided. Eng.*, vol. 14, pp. 187–212, 2007.

[33] S. McKennoch, D. L. D. Liu, and L. G. Bushnell, "Fast Modifications of the SpikeProp Algorithm," in *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, 2006, pp. 3970–3977.

[34] R. Gütig and H. Sompolinsky, "The tempotron: a neuron that learns spike timing-based decisions.," *Nat. Neurosci.*, vol. 9, no. 3, pp. 420–428, 2006.

[35] A. Vijayan, C. Medini, H. Singanamala, C. Nutakki, B. Nair, and S. Diwakar, "Classification of Robotic Arm Movement using SVM and Naïve Bayes Classifiers," in *Proceedings of Third International Conference on Innovative Computing Technology (INTECH 2013)*, 2013, pp. 263–268.

[36] S. Diwakar, J. Magistretti, M. Goldfarb, G. Naldi, and E. D'Angelo, "Axonal Na+ channels ensure fast spike activation and back-propagation in cerebellar granule cells," *J. Neurophysiol.*, vol. 101, no. 2, pp. 519–532, 2009.

[37] L. R. Rabiner, "Techniques for Designing Finite-Duration Impulse-Response Digital Filters," 1971.

[38] J. del R. Millán, "Brain-computer interfaces," *Handb. Brain theory neural networks*, vol. 110, pp. 67–74, 2013.