frontiers | Frontiers in Neuroscience

TYPE Original Research
PUBLISHED 28 November 2022
DOI 10.3389/fnins.2022.909146

Check for updates

OPEN ACCESS

EDITED BY
Jonathan Mapelli,
University of Modena and Reggio Emilia, Italy

REVIEWED BY
Claudia Casellato,
The University of Pavia, Italy
Michele Folgheraiter,
Nazarbayev University, Kazakhstan

*CORRESPONDENCE
Shyam Diwakar
shyam@amrita.edu

SPECIALTY SECTION
This article was submitted to Neural Technology, a section of the journal Frontiers in Neuroscience

RECEIVED 31 March 2022
ACCEPTED 02 November 2022
PUBLISHED 28 November 2022

CITATION
Vijayan A and Diwakar S (2022) A cerebellum inspired spiking neural network as a multi-model for pattern classification and robotic trajectory prediction.
Front. Neurosci. 16:909146.
doi: 10.3389/fnins.2022.909146

A cerebellum inspired spiking neural network as a multi-model for pattern classification and robotic trajectory prediction

Asha Vijayan[1,2] and Shyam Diwakar[1,3]*

[1]Amrita Mind Brain Center, Amrita Vishwa Vidyapeetham, Kollam, India, [2]School of Biotechnology, Amrita Vishwa Vidyapeetham, Kollam, India, [3]Department of Electronics and Communication, Amrita School of Engineering, Amrita Vishwa Vidyapeetham, Kollam, India

Spiking neural networks were introduced to understand spatiotemporal information processing in neurons and have found their application in pattern encoding, data discrimination, and classification. Bioinspired network architectures are considered for event-driven tasks, and scientists have looked at different theories based on the architecture and functioning. Motor tasks, for example, have networks inspired by cerebellar architecture where the granular layer recodes sparse representations of the mossy fiber (MF) inputs and has more roles in motor learning. Using abstractions from cerebellar connections and learning rules of deep learning network (DLN), patterns were discriminated within datasets, and the same algorithm was used for trajectory optimization. In the current work, a cerebellum-inspired spiking neural network with dynamics of cerebellar neurons and learning mechanisms attributed to the granular layer, Purkinje cell (PC) layer, and cerebellar nuclei interconnected by excitatory and inhibitory synapses was implemented. The model's pattern discrimination capability was tested for two tasks on standard machine learning (ML) datasets and on following a trajectory of a low-cost sensor-free robotic articulator. Tuned for supervised learning, the pattern classification capability of the cerebellum-inspired network algorithm has produced more generalized models than data-specific precision models on smaller training datasets. The model showed an accuracy of 72%, which was comparable to standard ML algorithms, such as MLP (78%), Dl4jMlpClassifier (64%), RBFNetwork (71.4%), and libSVM-linear (85.7%). The cerebellar model increased the network's capability and decreased storage, augmenting faster computations. Additionally, the network model could also implicitly reconstruct the trajectory of a 6-degree of freedom (DOF) robotic arm with a low error rate by reconstructing the kinematic parameters. The variability between the actual and predicted trajectory points was noted to be ± 3 cm (while moving to a position in a cuboid space of 25 × 30 × 40 cm). Although a few known learning rules were implemented among known types of plasticity in the cerebellum, the network model showed a generalized processing capability for a range of signals, modulating the data through the interconnected neural populations. In addition to potential use on sensor-free or feed-forward based controllers for robotic arms and as a generalized pattern classification algorithm, this model adds implications to motor learning theory.

KEYWORDS

cerebellum, spiking neural network, deep learning, encoding, classification, trajectory prediction

Introduction

The brain circuits of many animals have significantly improved their capacity to learn and process multimodal inputs (Stock et al., 2017) at the millisecond scale that machine learning (ML) algorithms have abstracted to classify (Albus, 1975) or cluster data (Kohonen, 1982). Yet, these algorithms are not as complex or efficient as the brain's neural circuits (Luo et al., 2019). Modern methods, such as deep learning networks (DLN) and extending artificial neural networks (ANN), may bring additional similarities to the computational capabilities of neural circuits while predicting and classifying patterns within big and small datasets (Angermueller et al., 2016), which could also help models to learn and think like humans (Lake et al., 2017). Studies involving DLN models suggest its application in deducing information processing within biological networks (Yamins and DiCarlo, 2016), in addition to disease-related predictions attributed to impaired network activity (Yang et al., 2014). Many DLN models have been inspired by the brain's microcircuit architectures, such as the visual cortex (Kindel et al., 2019), basal ganglia (Hajj and Awad, 2018), and hippocampus (Fontana, 2017), and these neuro-inspired models explore novel functional relationships within data.

Spiking neural networks (SNN) exploit a biologically observed phenomenological element in ML, allowing optimization and parallelizability to algorithms (Naveros et al., 2017) which may be event-driven and time-driven and may incorporate spatiotemporal information processing capabilities of biological neural circuits. Algorithms that are based on different brain circuits, such as the visual cortex (Fu et al., 2012; Yamins and DiCarlo, 2016), basal ganglia (Doya, 2000; Baladron and Hamker, 2015; Girard et al., 2020), and cerebellum (Casellato et al., 2012; Garrido et al., 2013; Antonietti et al., 2015; D'Angelo et al., 2016b; Luque et al., 2016; Yamaura et al., 2020; Kuriyama et al., 2021) with spiking neural models help understand the circuitry and in turn, help reconstruct and train systems. EDLUT (Ros et al., 2006), SpiNNaker (Khan et al., 2008), MuSpiNN (Ghosh-Dastidar and Adeli, 2009), and biCNN (Pinzon-Morales and Hirata, 2013) are some of the existing brain-inspired models which are used in the field of control systems for robotic articulation control. In this article, we mathematically reconstructed a cerebellum-inspired neural circuit incorporating the training efficacy of a deep learning classifier for applications in motor articulation control and pattern classification.

Among the different brain regions, the cerebellum, situated inferior to the occipital lobe of the cerebral cortex, has a modular structure and shares a common architecture (Standring, 2016) known to be involved with learning at different layers. Cerebellum follows a well-organized network structure (**Figure 1A**) with afferent circuitry involving two primary excitatory inputs to the cerebellum originating through the mossy fibers (MF) and climbing fibers (CF). Sensory and tactile inputs from different regions, including the brain stem and the spinal cord, are transferred by MF as input signals. At the same time, CF has been known to provide training errors compared with inputs from the inferior olive (IO) (Mauk et al., 1986). IO receives input from deep cerebellar nuclei (DCN), which are inhibited by the Purkinje neurons and considered to be the main output neurons of the cerebellar cortex (Eccles, 1973; D'Angelo et al., 2016a). The cerebellum granular layer has a large number of granule cells (GrC), significantly lesser numbers of Golgi cells (GoC), and a few Unipolar Brush Cells (UBC), while the primary neurons of the molecular layer are the Purkinje cells (PC). Due to the error correction mechanism by the IO onto the PC, the cerebellum has been known to perform a supervised motor learning of sensory and movement patterns (Porrill et al., 2004; Passot et al., 2010), although the relationship between the cerebellum and motor learning was suggested even earlier (Cajal, 1911). On the other hand, GrCs have been known to involve in sparse recoding of the different sensory modalities. The implementation of inhibition-based looping in the network model suggests that feed-forward inhibition of GrC may be crucial in modulating the efficacy of pattern discrimination of the PC, which is attributed to some of the biophysical characteristics of the PF-PC synapses. Generalization and learning-induced accuracy (Clopath et al., 2012) in the network model involves clustered activation (Diwakar et al., 2011) and synchronous behavior (Maex and De Schutter, 1998) in the GrC.

Sensorimotor control in the brain is often attributed to the cerebellum (Kawato and Gomi, 1992; Schweighofer et al., 1998; Thach, 1998; Ito, 2002; Nowak et al., 2007; Haith and Vijayakumar, 2009; Popa et al., 2012), which has been known to play roles in the error correction based control (Tseng et al., 2007; Shadmehr et al., 2010; Luque et al., 2011; Popa et al., 2016)

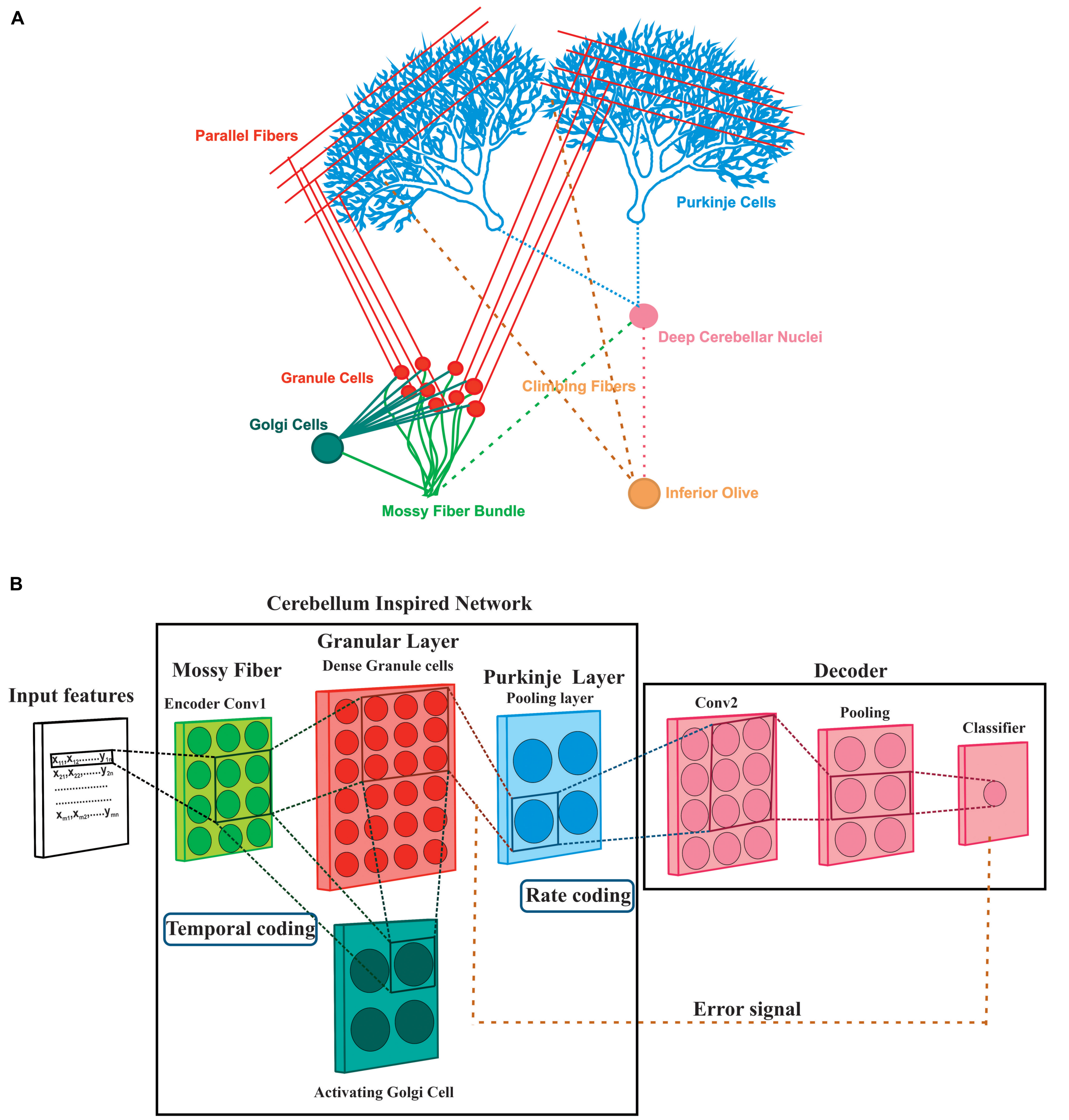

FIGURE 1

A deep layer network of the cerebellum. **(A)** The abstracted model was based on the simplified cerebellar microcircuitry of rat cerebellum: Various neurons and connections in the cerebellar circuitry may attribute to pattern classification and prediction. Input patterns are presented through the Mossy fibers (MF) which have projections on the Granule cells (Grc) extended as Parallel fibers (PF) and Golgi cells (GoC). PF impinges onto the Purkinje cells (PC) which gives input to the output neuron Deep Cerebellar Nuclei (DCN) which in turn evokes Inferior Olive (IO). From a deep learner perspective, the cerebellum has been known to perform supervised learning with the teaching signals send to the PC through the IO extension Climbing fibers (CF). **(B)** The equivalent model of deep learning involved a spiking neural network: All the neurons inside the cerebellum-inspired network were reconstructed as spiking models whereas the decoder was modeled with nodes. Each instance was mapped to a convolution layer that encodes to a group of MF. Convoluted cells mapped to a dense granular layer (Diwakar et al., 2011) representing the sparseness of the MF stimuli which is also activated by an inhibitory GoC model. Rate coding of the inputs, convolution at the GrC layer, and SoftMax based output prediction post the pooled layer was implemented at the MF-GrC-PC circuit.

by timing and coordinating movement (D'Angelo and Zeeuw, 2008). Studies have shown that the brain-inspired NN with a feed-forward and feedback controller could control a visually guided robotic arm (Schweighofer and Arbib, 1998; Tseng et al., 2007), saccades (Schweighofer et al., 1996), smooth pursuit (Kettner et al., 1997), and eye blinking condition (Casellato et al., 2014). Multiple controllers have several copies of inverse and forward models that could be coupled together to attain the task of fast and distributed coordination (Wolpert and Kawato, 1998; Kawato, 1999) and have been mainly used in unsupervised (Schweighofer et al., 2001) and supervised learning (Kawato et al., 2011) but not have had much focus on reinforcement learning (Yamazaki and Lennon, 2019; Kawato et al., 2020). Real-time processing of information for error detection and correction (Rodriguez-Fornells et al., 2002; Yamazaki and Igarashi, 2013; Luo et al., 2016; Popa et al., 2016; Yamazaki et al., 2019), predictive control (Kettner et al., 1997), the timing of muscle synergies (Manto et al., 2012), and formation of an internal model for supervised learning (Doya, 1999) as well as for calculating inverse dynamics (Kawato and Gomi, 1992) was attributed to the cerebellum because of their multi-faceted role in motor learning (Tanaka et al., 2020).

The cerebellum has similarities to DLN models with different learning modules and known supervised learning attributions (Eccles et al., 1967b; Marr, 1969; Albus, 1971; Ito, 1972; Doya, 2000). The memories formed can be stored in the cerebellar cortex and the deep nuclei. At the same time, the cerebellum elaborates over 16 known learning mechanisms (Mapelli et al., 2015), and the cerebellar cortex has been experimentally observed to be critical for regulating the timing of movements and learning is transferred partially or wholly to the deep nuclei. Plasticity at the MF-GrC connection (Nieus et al., 2006) has been less explored when compared with PF-PC plasticity. Olivo-cerebellar tract has also been seen to have a crucial role in learning new motor skills (Ito, 2006). Other hypotheses include the cerebellum as an adaptive filter (Fujita, 1982), a motor pattern classifier (Albus, 1975), and a neuronal timing machine that adapts to motor tasks (D'Angelo and Zeeuw, 2008). Cerebellum-inspired models have also been used in robotics, pattern separation, and ML applications (Ros et al., 2006; Dean et al., 2010; Tanaka et al., 2010; Casellato et al., 2014; Antonietti et al., 2015). Cerebellum-inspired network models have been attributed to several functions, such as vestibulo-ocular reflexes (VOR), optokinetic responses (OKR) (Inagaki and Hirata, 2017), eye blink classical conditioning (EBCC), and motor control with these different mechanisms attributed to the same neural circuitry relating a deep learning aspect into the cerebellum microcircuits (Wolpert et al., 1998; Hausknecht et al., 2016).

Sensory modalities evoke sparse activation of neurons where some neurons remain active all the time, whereas other neurons respond to few stimuli (Pehlevan and Sompolinsky, 2014) and this behavior has been observed in different parts of the brain, such as the cortex and cerebellum. Early studies (Marr, 1969; Albus, 1971) have suggested the sparse role of cerebellar granular layer neurons improves pattern separation. A measure of the sparseness or storage capacity of neurons was used to understand the responses of the network (Brunel et al., 2004; Pehlevan and Sompolinsky, 2014) at different sensory stimuli conditions in areas, such as the sensory cortex (Fiete et al., 2004), cerebellum (Clopath and Brunel, 2013; Babadi and Sompolinsky, 2014; Billings et al., 2014), etc. Models developed based on the cerebellum may be crucial for simulating movement changes in related disorders, such as spinocerebellar ataxia (Sausbier et al., 2004; Kemp et al., 2016) and multiple sclerosis (Redondo et al., 2015; Wilkins, 2017). As sparseness and capacity indicate the amount of computation and storage, these measures are now being used in modeling. They are also calculated in terms of bits of information processed (Varshney et al., 2006) which may be employed to assess DLNs. Here, we reconstructed a cerebellum-inspired spiking neural network with aspects of biophysical dynamics and some of the learning mechanisms of the cerebellum. The focus of this article was also to build a multipurpose algorithm that could perform tasks, such as pattern encoding, discrimination, separation, classification, and prediction critical to motor articulation control. The modeled network demonstrates capabilities attributed to a pattern classifier and functions implicitly as a trajectory predictor for motor articulation control resembling the sensorimotor control observed in the cerebellum. The simulated cerebellar network was analyzed for its generalization capability aiming for pattern recognition and supervised learning. Toward quantifying performance, the model implemented on CPUs was compared with implementations on graphics processing unit (GPUs) and against some of the other learning algorithms.

In the current study, we aim to reconstruct a cerebellum-inspired spiking neural network extended from a previous study (Vijayan et al., 2017) that could perform tasks with multiple configurations like pattern classification and trajectory prediction using the same network architecture. The network has been mathematically reconstructed for two aspects: an extension of the theory of cerebellar function (Marr, 1969; Albus, 1971; Ito, 1982) where we looked at employing the modern understanding of the cerebellar cortex and at the same time, develop a neuro-inspired bio-realistic approach for low-cost robotic control. Unlike in the previous model (Vijayan et al., 2017), which had a granular layer with only GrCs, we have incorporated the recoding and associative mapping properties of the cerebellar granular layer with excitatory GrC and an inhibitory GoC, pattern discrimination at the Purkinje layer, and the interpretational application of DCN. Moreover, the network size was scalable. Extending the Marr–Albus–Ito theory, we have included learning in the granular layer, suggesting there is a whole set of operations that are done by the granular layer and would be performed as indicated by some experimental studies (D'Angelo et al., 2001, 2013; Rössert et al., 2015). The reconstructed cerebellum-inspired spiking neural network model performed tasks of different mathematical capabilities of this connectivity and circuit and can be repurposed to help model disease conditions as well as in developing controller models for sensor-free neuroprosthesis.

Materials and methods

Neuronal models for spiking networks

A cerebellum-inspired spiking neural network model (**Figure 1B**) with deep learning functionality was developed, extending our previous implementation (Vijayan et al., 2017). The input stimuli (MF) and the cerebellar neurons, such as GrC, GoC, and PC, were reconstructed using the Adaptive Exponential Integrate-and-Fire (AdEx) model (Naud et al., 2008). The nine parameter values for the AdEx model were obtained using a PSO algorithm (**Table 1**). The firing behavior was reconstructed for a single neuron with specific current values (**Table 1**) and matched the experimental recordings from p17 to 23 rat cerebellum cerebellar neurons (D'Angelo et al., 2001) and validated with modeling articles (Medini et al., 2012, 2014). Spiking dynamics and the neuron's firing rate adaptation were modeled using Eqs (1)–(3).

$$\mathrm{C}\frac{d\mathrm{V}}{dt} = -g_L\left(V-E_L\right)+g_L\triangle_T\exp\left(\frac{V-V_T}{\triangle T}\right)+\mathrm{I}-w \qquad (1)$$

$$\tau_w\frac{dw}{dt} = \mathrm{a}\left(\mathrm{V}-E_L\right) \qquad (2)$$

$$If\ V > 0\,mV\ then\begin{cases} V=V_r \\ w=w+b \end{cases} \qquad (3)$$

Here, C was the membrane capacitance, g_L represented leak conductance, E_L denoted resting potential, Δ_T represented slope factor, and V_T denoted threshold potential. Variable "w" described the adaptation factor within the membrane potential, and "a" represented the relevance of sub-threshold adaptation. "I" referred to the injected current applied from an external source, and "b" referred to the spike-triggered adaptation constant.

The cerebellum-inspired deep learning algorithm included four component modules: an encoder to translate real-world data, a spiking cerebellar microcircuit-based network, a decoder of spiking information, and a learning and adaptation module to update weights at different layers in the algorithm (**Figure 2**).

TABLE 1 Parameter values used in modeling the different types of neurons.

Parameter	Values		
	GrC	GoC	PC
C (pF)	150	500	100
g_L (nS)	10	13.1	10
E_l (mV)	−70	−58	−65
Δ_T (mV)	4	7	2
V_t (mV)	−50	−54	−50
τ_w (mS)	13	8.7	1
I (pA)	350	0	0
a (nS)	9	−17	−13
b (pA)	250	1,033	260

Encoding of real-world data

Dataset[1] used as training input consisted of different features $[x_1,x_2,x_3...x_n] \in X$ with a class label of y. The input features were encoded (**Algorithm 1**) into spikes using a convoluted Gaussian kernel function. The attributes were mapped to a dataset-based variable in higher dimensional space, simulating the MF-GrC elaboration. Model currents were estimated as weighted input values while computing spikes from the encoded data through the granular layer. In the mapping of inputs, a normal distribution similar to postsynaptic latencies (Silver et al., 1996) allowed a center-surround structure (Mapelli et al., 2010) to the inputs (neuron in the center received the strongest excitation). The neurons were made user-defined and scalable. The convolution layer in DLN was designed to abstract the relevant features from a dataset. In the cerebellum, the MF have been known to bring relevant information from the higher centers (**Figure 1**).

Cerebellum-inspired spiking neural network

A scalable cerebellar network was modeled (**Algorithm 2**) consisting of MF, GrC, GoC, and PC models. One of the aspects of reconstructing a cerebellar network was to employ the modern understanding of the cerebellar cortex and to solve motor task classification and prediction. Each feature was represented as a set of *n* neurons. The number of MF inputs was set based on the dataset feature size. For a dataset with 4 features and with 7 neurons per feature (see **Supplementary Method 5** and **Supplementary Table 1**), the network consisted of 28 MFs, 371 GrCs, 1 GoC, and 1 PC. Twenty-eight MFs provided excitatory input to 371 GrCs and 1 GoC. An inhibitory GoC connection modulated all GrCs and its projection on the PC was considered as the output to the network. Convergence was set as in previous studies (Solinas et al., 2010; Vijayan et al., 2017). The weights were initialized using the standard approach of randomly picking from a normal distribution at each layer to avoid the explosion or vanishing of the activation layer output. The model was simulated for 300 ms. Each MF was mapped to 53 granule cells considering the convergence–divergence ratio (Eccles et al., 1967a). Every feature was mapped to 7 neurons, so for a 4-featured data point, there were 28 input neurons (MF) mapped to GrC and then to the PC. The output of the PC was decoded to the final output through 3 layered networks. The Purkinje layer was considered as the pooling layer where the output from the dense granular layer was summed up. The output from the pooled layer was sent to the second convolution layer, where the features are extracted, pooled, and classified (**Figure 1B**).

[1] https://github.com/compneuro/DLCISNN

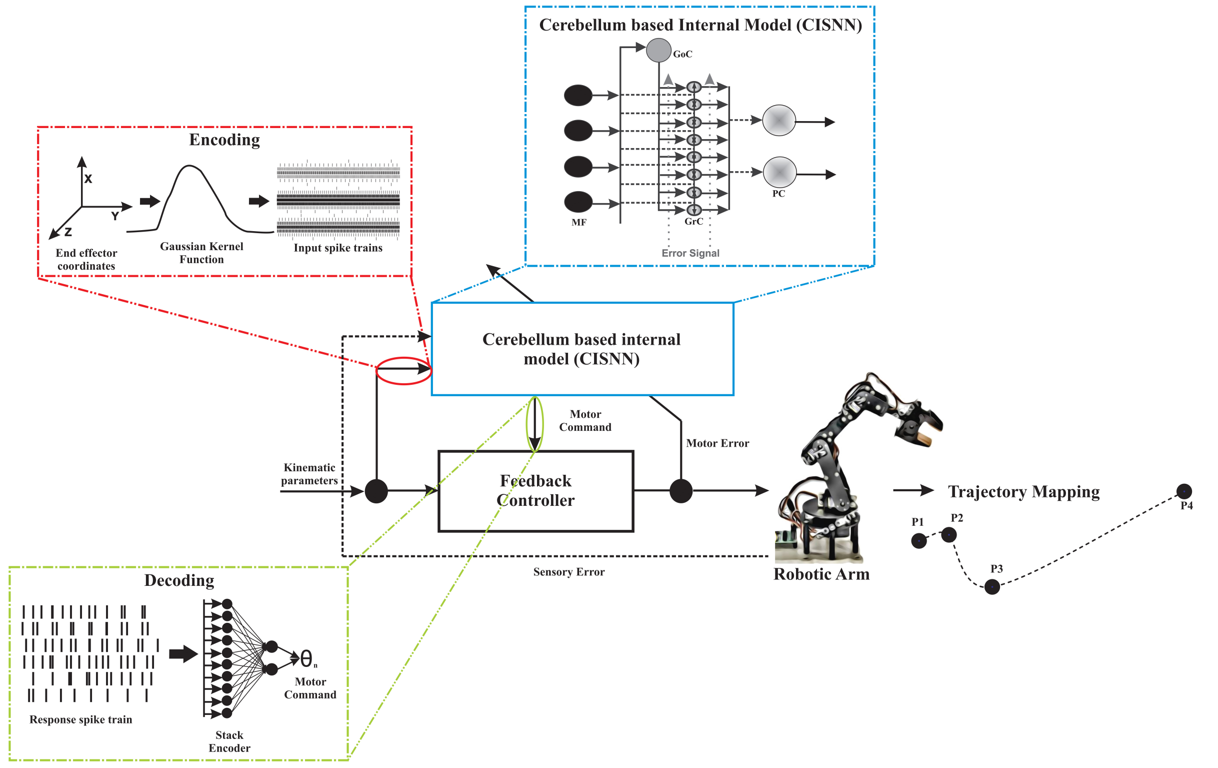

FIGURE 2
The network topology for robotic abstraction to predict trajectory. Input parameters (kinematic parameters) were encoded to MF spike times using a Gaussian kernel and are temporal coded. The temporal coded information is given as input to the cerebellum-inspired spiking neural network (CISNN). CISNN is a layered network with MF, GrC, and PC layers. The output of CISNN was obtained as PC spike times which were then rate coded by employing a stack encoder, the output obtained was presented to a low-cost robotic articulator that mapped the 4-point trajectory. Changes in sensory and motor values were calculated and fed back to the network to learn and update the existing weights.

Algorithm 1: MF-granular input encoding using a Gaussian Kernel based convolution.

```
Preprocess data X by normalizing the
features using Min-Max normalization
  $D_{norm} = \frac{x_i - \min(x)}{\max(x) - \min(x)} * (n_{max} - n_{min}) + n_{min}$
Where X is the dataset [$x_i$] ∈ X,
$n_{max}$ is the new maximum, $n_{min}$ is the
new minimum
For each instance i
    For each attribute j
      Compute $f(D_{norm}(i, j))$ pdf for using
      a normal distribution function
      Compute convolution matrix C
      by using a full convolution of
      the pdf obtained and $D_{norm}(i, j)$.
      Compute the scalable datapoint
      for each feature $D_j$ = [$\mu_{ij}$-3σ,
      $\mu_{ij}$-2σ, $\mu_{ij}$-σ, $\mu_{ij}$, $\mu_{ij}$+σ, $\mu_{ij}$+2σ,
      $\mu_{ij}$+3σ]
      Compute $w_j$ which was $f(\mu_{ij})$
      Check if C contains datapoints
        Compute I = $w_j * \mu_{ij}$
      End if
    End for
    For each D
      Create spiking neurons with
      the computed I as current
    End for
End for
```

Algorithm 2: Cerebellum inspired spiking neural network.

```
Create n MF neurons with $x_j$:$N_{MF} = n * x_j$
Create Granular layer
   Golgi cells GoC created based on
   the MF count $N_{MF}$
   Connection weights $w_{MF-GoC}$
   were set to a random range of
   values [0.01-0.09] of size
   $N_{MF}$ x $N_{GoC}$
```

```
    Where $N_{GoC}$ is the number of Golgi cells
    For each GoC
        For each MF
            Compute $I_{GoC}$ by adding $w_{MF-GoC}$ (MF,GoC) at time $T_{MF}$
            Where $T_{MF}$ is the Mossy fiber spike time
        End for
        Create Golgi neuron models with the computed $I_{GoC}$ using Eq. (1)-(3)
    End for
    Compute $N_{GrC} = \frac{(N_{MF}*53)}{4}$
    Granule cells GrC created based on $N_{MF}$, $T_{MF}$, $N_{GoC}$, $T_{GoC}$
    Where $T_{GoC}$ is the Golgi cell spike timings
    Connection weights $w_{MF-GrC}$ were set to a uniform random value of size $N_{MF}$ x $N_{GrC}$
    Connection weights $w_{GoC-GrC}$ were set to a random range of values [0.01-0.09] of size $N_{GoC}$ x $N_{GrC}$ were set
    Number of activated MF, $A_{MF}$, for each GrC was set to 4
    For each GrC
        For each $A_{MF}$
            Compute $I_{GrC} = I_{GrC} + w_{MF-GrC}$ ($A_{MF}$,GrC) at $T_{MF}$
        End for
        For each GoC
            Compute $I_{GrC} = I_{GrC} - w_{GoC-GrC}$ (GoC,GrC) at Golgi cell spike time $T_{GoC}$
        End for
        Create Granule neuron model with the computed $I_{GrC}$ and the spike train using Eq. (4)
    End for
Purkinje cells PC was created based on the number of output features $y_j$; $N_{PC}$ = count ($y_j$)
Connection weights $w_{GrC-PC}$ were set to a value of 0.01 of size $N_{GrC}$ x $N_{PC}$
Number of activated GrC, $A_{GrC}$, for each PC was set as 48
For each PC
    For each $A_{GrC}$
        Compute $I_{PC} = f(T_{GrC}) * w_{GrC-PC}$ ($A_{GrC}$,PC) and the spike train using Eq. (7)
    End for
        Create Purkinje neuron model with the computed $I_{PC}$ using Eq. (1)-(3)
End for
```

The spike trains for each neuron in the granule layer was defined by Eq. (4)

$$PSP_{Grc} = w^T x_{MF}(t) = \sum w_{MF} x_{MF}(t) \tag{4}$$

Where w^T was the complete weight matrix for the MF-GrC connection in which x_{MF} was calculated using Eq. (5)

$$x_{MF}(t) = \sum_{t^f < t} \delta(t - t^f) \tag{5}$$

The input spike times is represented by t^f and $\delta(x)$ is a dirac delta function which follows

$$\delta(x) = \begin{cases} 1 & ifx = 0 \\ 0 & otherwise \end{cases} \tag{6}$$

The same mathematical formulation was applied to the Purkinje layer Eq. (7)

$$PSP_{PC} = w^T x_{Grc}(t) = \sum w_{Grc} x_{Grc}(t) \tag{7}$$

During adaptation and learning, each of the spike train sequences were convolved with a kernel function Eq. (8)–(9)

$$\tilde{x}_{MF}(t) = \sum_{t^f_{MF} \in F_{MF}} \kappa(t - t^f_{MF}) \tag{8}$$

$$\tilde{x}_{Grc}(t) = \sum_{t^f_{Grc} \in F_{Grc}} \kappa(t - t^f_{Grc}) \tag{9}$$

Where F_{MF} and F_{GrC} were the set of spike trains inputs to the respective granular and Purkinje cell layers.

Decoding spiking information for classification

A decoder block follows the spiking cerebellar network to decode the firing behavior to real-world information. The decoding network consisted of an input layer with nodes based on the time bin, a hidden layer with two nodes and an output node. The output was decoded using rate-coded information from the Purkinje cell spike patterns and a convoluted neural network employing a SoftMax classifier (Algorithm 3). Rate coding was performed by dividing the time interval into time bins of 50 ms and counting the number of spikes in each bin. Spike count was stored as a vector and was used as the input layer of the convolution neural network. Values for all nodes were computed using a weighted sum of inputs.

Learning rules and adaptation

Optimization of the network was done by correcting the network connection weights with calculated errors. The predicted output was compared with the actual to compute errors. During the training phase, connection weights were updated at the granular layer ($\mathbf{w}_{MF-GrC}$), Purkinje layer ($\mathbf{w}_{GrC-PC}$), and the decoding network layer (Algorithm 4).

Algorithm 3: Decoding using a rate coded convoluted network.

```
1. For each output feature y
2.     Divide the time interval T into B
       time bins of time t : T = B*t
3.     Count the number of spikes in
       each bin as vector
       $C_s = \{n_1, n_2 .... n_B\}$
4.     Assign [$C_s$] set as input layer
       and No. nodes j = length($C_s$)
5.     Assign No. Hidden layers i = 1;
       and No. hidden nodes j = 2
6.     Assign No. of output node j = 1
7.     Assign weights w were set at each
       layer of size j * i
8.     For each layer i
9.         For each node j
10.            Compute $s_j = \sum w_{j,i} x_i + b_i$
11.            Compute $f(s_j) = \frac{1}{1+e^{-s_j}}$
12.        End for
13.    End for
14.    Compute output node $s = \sum w*x+b$
15.    Compute $f(s) = \frac{1}{1+e^{-s}}$
16.    Check if $f(s) \le 0.5$
17.      Assign y' = 1
18.    Else
19.      Assign y' = 0
20.    End if
21.    End for
```

Algorithm 4: Learnadapt for network learning and weight adaptation.

```
1. Compute the error e = y-y'
   Where y is the actual output and y'
   is the predicted output
2. Compute the adapted weight for
   $w_{MF-GrC}$
3. For each MF
4.     For each GrC
5.         Assign learning rate $l_r$ = 0.1
6.         Compute $w_{MF-GrC}(n+1) = w_{MF-GrC}(n)$
           $-(l_r*e* w_{MF-GrC}(n))$
7.     End for
8. End for
9. Compute the adapted weight for
   $w_{GrC-PC}$
10. For each GrC
11.     For each PC
12.         Assign $l_r$ = 0.35
13.         Compute
            $w_{GrC-PC}(n+1) = w_{GrC-PC}(n)$
            $-(l_r*e*w_{GrC-PC}(n))$
14.     End for
15. End for
16. Compute adapted weight at Rate
    Coded Convoluted network
17. Compute error signal
    $e_s = (y-y')*y'*(1-y')$
18. Assign $l_r$ = 0.15
19. Compute $w(n+1) = w(n)-(l_r*e_s*s(n))$
20. For each hidden layer i
21.     For each hidden node j
22.         Assign $y' = s_j$
23.         For each previous layer
            node k
24.           Compute $\triangle e = \sum e_{s_j}(i+1)*w(i)_{jk}$
25.         End for
26.         Compute $e_{s_j}(i) = \triangle e*y'*(1-y')$
27.         Check if i = 1
28.           Compute
              $w_{ji}(n+1)=w_{ji}(n)-(l_r*e_s*x_j(n))$
29.         Else
30.           Compute
              $w_{ji}(n+1)=w_{ji}(n)-(l_r*e_s*s_{j-1}(n))$
31.         End for
32.         End for
```

The learning rule for updating synaptic weights was employed as in Eqs (10)–(12)

$$\Delta w_{Grc}(t) = \lambda \tilde{x}_{Grc}(t)\, e_m \tag{10}$$

$$\Delta w_{MF}(t) = \lambda \tilde{x}_{MF}(t)\, e_s \tag{11}$$

Where λ was the learning rate, e_s and e_m were the sensory error and motor error, respectively, used for the prediction task. For the classification task, a single error representation was used (e_s and e_m errors were considered the same) and was calculated using the general form,

$$e(t) = (y_d(t) - y_{out}(t)) \tag{12}$$

Here, y_d was the desired motor or sensory commands and y_{out} was the predicted.

The weights across the network were modified (10.11) based on the architecture of the modeled network and were critical to the changes observed in the input layer of the model as observed in deep convolution networks (Diehl et al., 2015; Pfeiffer and Pfeil, 2018). Only a small portion of the granule cells was turned silent, and this sudden sparseness itself was important for the learning process (Galliano et al., 2013). Bidirectional plasticity was introduced at both the synapses, with the weight at MF-GrC and GrC-PC being updated based on a Hebbian-like learning rule (error instead of inputs), and weights at the decoding network were updated based on a modified backpropagation learning rule. Optimized learning rates were used for the study, which was obtained from an extensive trial-and-error method (see **Supplementary Method 3**) based on the rate of cerebellar learning at various synapses, with PF-PC having the fastest learning to the PC-DCN with the slowest learning (Schweighofer and Arbib, 1998; D'Angelo et al., 2016b; Herzfeld et al., 2020). For the cerebellar network, the learning rates for the algorithm were kept as low as 0.1, 0.35, and 0.15 [optimal choice based on trial-and-error (see **Supplementary Method 3** and **Supplementary Figure 3**)] for the MF-GrC, PF-PC, and PC-DCN connections, respectively.

The MATLAB (Mathworks, USA) source code for the cerebellar network for classification is available at https://github.com/compneuro/DLCISNN and will be made available on ModelDB.

Dataset

To train and validate this model (**Algorithm 5**), various datasets from the UCI repository (Lichman, 2013) were used. The algorithm was tested on autism spectrum disorder (ASD) for children, adolescents and adults dataset (Thabtah, 2017), Iris dataset, play tennis (Soman et al., 2005), and robotic arm dataset (Vijayan et al., 2013). All the 3 ASD datasets had 19 features with more than 100 instances, iris and play tennis had 4 features, and the robotic arm dataset had 9 features. WEKA data mining platform (Witten et al., 2011) and the deep cognition platform (deepcognition.ai) were used for comparing the cerebellum model with other ML algorithms.

Algorithm 5: CISNN training and testing.

```
1. Get the dataset $D_T$ from the user
2. Split $D_T$ into training ($D_{Tr}$) and
   test data ($D_{Ts}$) using 66% split
3. For each $D_{Tr}$
4.     Encoding the realworld data
       $T_{mfspiketime}$ = MF-Granular input
       Encoding ($D_{Tr}$)
5.     Create spiking neural network
       $T_{pcspiketime}$ = Cerebellum Inspired
       Spiking Neural Network ($T_{mfspiketime}$)
6.     Decode the spiking information to
       output $y'$ = Decoding ($T_{pcspiketime}$)
7.     Compute the error $e = y-y'$
8. End for
9. For each epoch
10.    Update network weights using
       Learnadapt(e)
11. End for
12. For each $D_{Ts}$
13.    Encoding the realworld data
       $T_{mfspiketime}$ = MF-Granular input
       Encoding ($D_{Ts}$)
14.    Create spiking neural network
       $T_{pcspiketime}$ = Cerebellum Inspired
       Spiking Neural Network ($T_{mfspiketime}$)
15.    Decode the spiking information to
       output $y'$ = Decoding ($T_{pcspiketime}$)
16.    End for
```

Modeling motor articulation control

A modified version of the cerebellum-inspired spiking neural network (Vijayan et al., 2017; **Figure 2**) was also used to predict the inverse kinematics of a simple sensor-free robotic articulator. However, inverse kinematics was ill-posed concerning the solution's existence, uniqueness, and stability, and solving this required multilayer networks with different regularization functions (Ogawa and Kanada, 2010). The current model was designed to solve these problems. The model input features were the kinematic parameters which were end-effector coordinates or motor values or both, that were provided to the network model to predict the output in the form of class labels or motor angles. Both input and output were decided based on the task. A 6-degree-of-freedom (DOF) robotic arm dataset generated from the robotic arm was used to test the predicted result (Nutakki et al., 2016; also see **Supplementary Figure 1**). In this case, the cerebellar network model for prediction consisted of 21 MF, 279 GrC, 1 GoC, and 6 PC for encoding seven features per neuron. The neurons were scaled up for different tasks based on the input features and user-defined MF count (see **Supplementary Method 5** and **Supplementary Table 1**). As the cerebellum is known for error-driven motor learning, a dual error representation (sensory error and motor error) was considered for the error correction (Popa et al., 2016). Decoded values were used to calculate the two types of errors: sensory error and motor error. The motor error was considered as the difference between the desired and the predicted motor angle, while sensory error represented the change in end-effector coordinates defined in the Cartesian space. A geometric estimate was used to calculate the end-effector coordinates from predicted motor angles, and this change in end-effector coordinates was considered as a sensory error. The sensory error feedback was to minimize the error, which biologically would have been done by the other loops in the circuit (Doya, 2000). In the case of motor error, it represents the error signal through the CF to the PC. Learning at the different layers has been designed based on the two types of errors. The sensory error was used to update the weight at the MF-GrC and the motor error at the GrC-PC and the layers in the decoder. Normalization functions were used to optimize the predicted joint angles, which was considered a regularization function. The predicted values were normalized using a min-max normalization function from a 0–1 range to 0–4.45, which corresponded to the angle in radians.

Sparseness and capacity measures

The sparseness index (SI) (Tartaglia et al., 2015) of the granular layer cells was calculated to measure the sparseness involved in the coding scheme at the granular layer. The SI was calculated as in Eq. (13).

$$SI = \frac{1 - A}{1 - \frac{1}{n}} \tag{13}$$

Where A was calculated as in Eq. (14) and n is the number of stimuli.

$$A = \frac{\left(\sum_1^n v_i/n\right)^2}{\sum_1^n \left({v_i}^2/n\right)} \tag{14}$$

Where v_i represents the mean firing rates to a set of stimuli, and SI can take only values between 0 and 1.

$$SI = \begin{cases} 0 & when\ v_i = v\ for\ all\ i \quad (A = 1) \\ 1 & when\ v_i = v\ and\ v_{j \neq i} = 0 \quad (A = \frac{1}{n}) \end{cases} \tag{15}$$

Analytically, it was difficult to estimate the number of input-output associations that can be stored in a neuron as it depended on the weights' dimensionality and the number of labeled patterns. Extensive computer simulations were required to calculate the number of input-output associations stored in a neuron (Memmesheimer et al., 2014). The storage capacity (α_c) of the Purkinje cells (Rubin et al., 2010; Clopath et al., 2012; Vijayan et al., 2015) in the network was also calculated to quantify the number of input-output associations that can be stored. The storage capacity with respect to stability constant (κ) and mean rate of output spike time ($r_{out}\ \tau$) was calculated Eqs. (17) (18) based on the duration of input pattern (T) for both the granule cell and Purkinje cell and the number of active MFs (N_τ) in a particular time window. The stability constant of the neuron was estimated using the Eq. (16) (Rubin et al., 2010), where τ_s and τ_m represent the synaptic time constant and membrane time constant.

$$\kappa = \frac{T}{\sqrt{\tau_s \tau_m}} \tag{16}$$

$$\alpha_c = \frac{ln ln \kappa}{2 ln 2} \tag{17}$$

$$\frac{T}{N_\tau} = \alpha_c (r_{out} \tau) \tag{18}$$

The membrane time constant (τ_m) for the Purkinje cell was taken as 64 ± 17 ms (Roth and Hausser, 2001) and the granule cell was taken as 1.4 ± 0.12 ms (Delvendahl et al., 2015). The synaptic time constant (τ_s) was calculated as $\tau_m/4$ (Rubin et al., 2010). When MF inputs were provided, the granule cells had a r_{out} in the range 10–50 Hz (Gabbiani et al., 1994; D'Angelo et al., 2001). In the case of PC, which was a spontaneous firing neuron model has a r_{out} of 30 Hz, and as PF impinged onto the PC dendrites, the r_{out} had a range of 30–500 Hz (Khaliq and Raman, 2005; Masoli et al., 2015). For the theoretical calculation of storage capacity, two inequalities exist; in the case of GrC, the $r_{out}\tau < = 0.1$ was used and for PC $r_{out}\tau > = 1$.

Graphics processing unit implementation of the cerebellar input layer model

To compare the computational costs during scaling, a modified version of the network implementation on the GPU platform was employed (see **Supplementary Figure 2**). Several network modules resembling the cerebellar microzones were reconstructed by maintaining the realistic neural density as reported in Solinas et al. (2010). Since granule neurons were more numerous than the other types of neurons reconstructed in the cerebellar cortex and the computations in the granular layer were embarrassingly parallel, GPU was used as a candidate to parallelly execute the cerebellar granular layer neurons. For GPGPU simulations, data-parallel processing mapped data elements to parallel processing threads, and the memory requirements for GPU processes were calculated at runtime, suggesting our model was automatically scalable in terms of both computational units and memory requirements (Nair et al., 2014).

Results

Cerebellar circuit operates as a deep learning neuronal machine

The encoding algorithm attributed spike timing and precision as a data-dependent coding feature. Reliability of input transformation has been associated with the efficacy of spike generation attributed to MF-GrC relays as well as spike precision and timing (Cathala et al., 2003). The initial convolution layers represented transformation by the granular layer followed by aggregations at the Purkinje layer representing the bidirectional plasticity modulated by PF-PC input convergence (Jörntell and Hansel, 2006). For classification and trajectory prediction-based studies, the number of spiking neurons in the input layer and the convolution layer increased with an increase in the number of features. Each feature set was encoded as an input pattern of spike trains. The number of MF was chosen at the user's discretion, and in the current model, seven neurons were used as they could represent the center-surround spread of data. The input data were mapped to a higher hyperspace allowing seven different sets of patterns for each feature in an instance, suggesting the cerebellum input layer mapping modalities involve input transformations like those reported for sensory tasks, such as VOR and EBCC (Medina and Mauk, 2000).

An inhibitory Golgi behavior was modeled to provide inhibitory input to the network, and it was observed that when an inhibitory spike train was introduced to the network, the granular layer showed a synchronous behavior (**Figure 3A**). From the average firing rate of the GrCs, it was also observed that minimal learning happens at the MF-GrC (**Figures 3B,C**) when compared with the PF-PC synapse. After learning, it was also observed that some of the granule cells turned silent (**Figure 3A**). There were also cases wherein after the initial iteration there were granule cells that were in a quiescent state and as learning progresses these neurons tend to fire and other neurons tend to change to a quiescent state. Learning at the different layers was representative of the plasticity-based learning at the MF-GrC, PF-PC, and PC-DCN. PC-DCN in the model was a transformation function that has error-based learning. It was observed that for a classification-related study when the learning rate was increased from 0.01 to 0.23 at the PC-DCN synaptic model, the training was fine-tuned at less than 5 iterations, thus mimicking the cerebellar fine-tuning behavior.

Cerebellum-based model noisily encodes input patterns

The degree of sparseness for granule layers neurons was analyzed using the SI measure. The SI depended on the firing rate based on the stimulus, and the mean SI for the granular layer neurons was observed to be 0.9855 ± 0.0625 before the training and 0.6276 ± 0.22 after the training (**Figure 4A**). As the SI increased, the signal-to-noise ratio increased, thus increasing the memory capacity, thereby decreasing classification error (Babadi and Sompolinsky, 2014; Rössert et al., 2015).

After the training, the PC spikes increase and tend to increase the firing average also (**Figures 5A,B**). The storage capacity (or the length of sequence that could be learned) of the modeled neurons was analyzed to understand the neurons' ability to store stimuli-response associations. The mean output spike times for the PC range from 30 to 500 Hz (Khaliq and Raman, 2005; Masoli et al., 2015) and with an increase in firing frequency, storage capacity decreased (**Figure 5C**). For the granule cells whose mean output spike times range from 10 to 50 Hz (Gabbiani et al., 1994; D'Angelo et al., 2001), the capacity of the cell was found to reduce with an increase in the firing frequency (**Figure 4C**). The reliability of the neuronal models was calculated from the storage capacity (**Figures 4B, 5D**). As the reliability factor (k) increased, the firing frequency also increased without increasing the information capacity of the neuron. It was observed that the PC showed a higher storage capacity compared with the GrC. The PC was attributed to an increase in storage capacity (Varshney et al., 2006). A comparison study (Vijayan et al., 2015) was previously carried out on the storage capacity estimated and indicated a similar result of ∼0.2–0.3.

Comparison with other machine learning algorithms

ML algorithms, LibSVM, MLP classifier, Dl4jMlp classifier, and RBF network were used to compare the performance of the cerebellum-inspired neural network model (**Figure 6A**). The cerebellar network showed an increase in accuracy with training epochs when the learning rate at PC-DCN was less than 0.01 and saturated slowly compared with the other ML methods. It was also observed that as the learning rate at PC-DCN increases from 0.01 to 0.23 accuracy reaches optimum by less than 5 epochs which is one of the features of DLNs. Although less efficient on small datasets, deep learning algorithms show 100% accuracy, which may be attributed to rote learning. Among the classifiers, the Dl4jMlpClassifier, which resembled the cerebellar model had a higher test efficiency of 96.96% for ASD_adolescent dataset, 95.6% for ASD_adult dataset, 100% for ASD_adult dataset, 78.4% for Iris dataset, 80% for play_tennis, and 64.28% for robotic arm data.

Feature generalization capability was input dependent

The model was used to evaluate the classification accuracy on different ML datasets using a 66% split. To test the generalization capabilities of the cerebellum-like spiking network model, various ML datasets were used, and the number of instances involved in the training process was: 131 (ASD_adult), 65 (ASD_adolescent), 164 (ASD_child), 99 (Iris), 27 (robotic arm), and 10 (play_tennis). Training time and test time increased with an increase in the number of features as the network formed was based on the number of input features (**Figures 6B,C**). Since input organization was not strictly connected to a layered set of neurons, the complexity of a feature in relation to all other features seemed more relevant while recruiting a dataset-driven architecture. Scaling to very large datasets, required parallelization of the inputs layer geometry reducing the time complexity associated with the network. In this direction, we attempted a simpler input code as volume geometry without adaptation (**Supplementary Figure 2**).

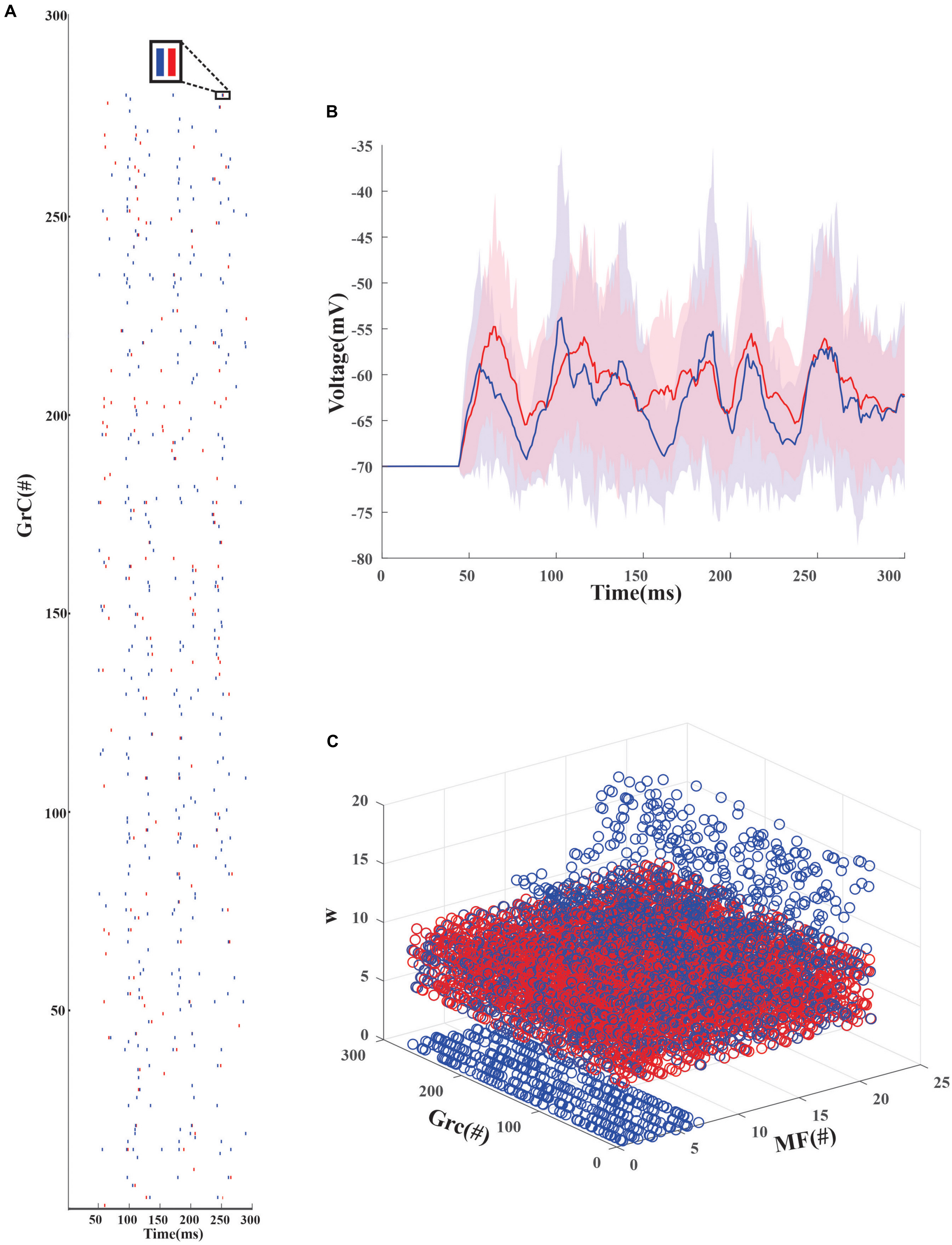

FIGURE 3
Firing rate distribution encoded input data in the modeled granule cells in the network. **(A)** Firing dynamics of GrCs pre and post-learning for 300 ms show the spread of temporal data in the input granular layer resulting from a combination of 4 random MF connections. The red dots represent the spike times of 279 GrC pre-learning and the blue dots represent the updated spike times post-learning. The inset shows the change in the firing dynamics at a particular time point for 1 GrC. Post-learning, some of the granule cells which were firing (e.g., only red raster in GrC1) turn quiescent. **(B)** Average voltage traces of granule neurons with standard deviation showed minimal change in the firing dynamics before (red) and after (blue) learning **(C)**. Weights before (red) and after (blue) learning suggest, post learning some GrCs adapt the weights to remain in a state of non-firing.

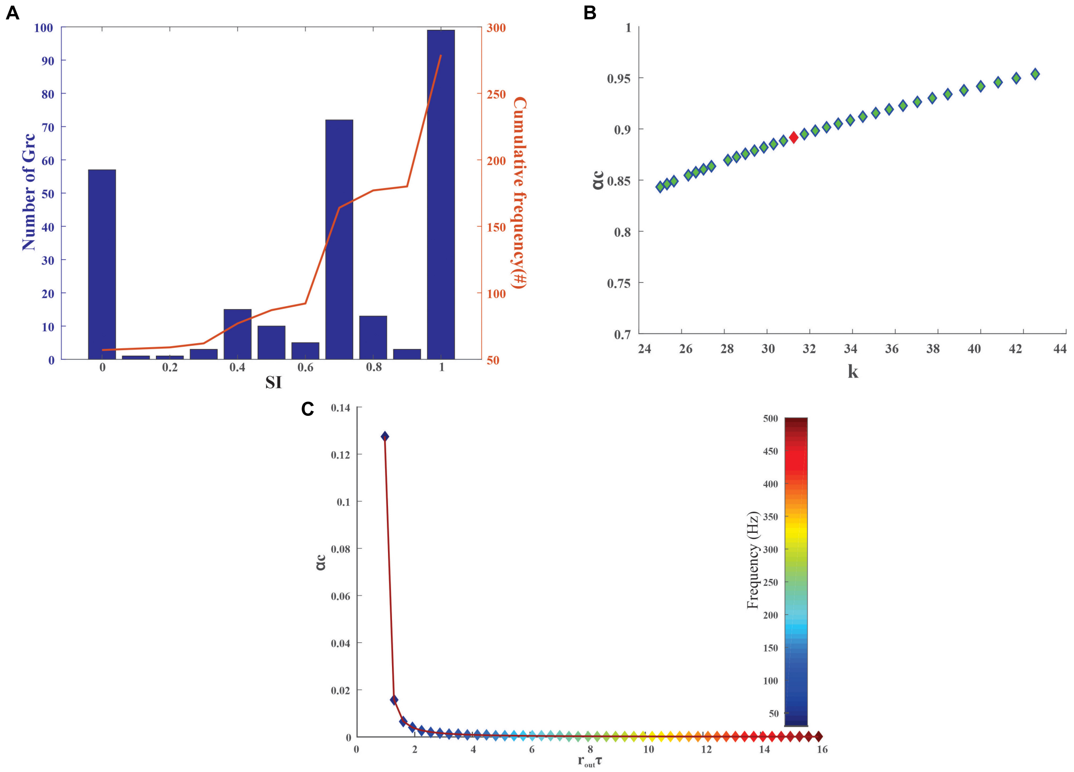

FIGURE 4
Quantification of stored information in the granule layer neurons of the cerebellum-inspired model. **(A)** Granular layer encoding relied on connection sparseness. The sparseness index of 279 granule neurons in the network with the encoded stimuli as input was calculated after 50 epochs. As iterations increase the granule cells tend to be sparser (SI nearing 1) as seen from the cumulative frequency. **(B)** Storage capacity (α_c) of granule neurons with respect to stability constant (κ) represents the formation and storage of new patterns. The red dot represents α_c for a GrC whose membrane time constant (τ_m) was 1.4 ± 0.12 ms. **(C)** Storage capacity (α_c) of GrC with a firing frequency of 10–50 Hz was calculated and as the mean firing output spike time ($r_{out}\tau$) increases, the storage capacity decreases steeply suggesting that a reduction in firing frequency leads to an increase in pattern storage.

With the granular layer computations being inherently parallel, we compared the implementation efficacy of 3.5 million granule neurons (**Supplementary Figure 2**). Large-scale implementations were feasible on GPGPUs with code parallelization. A smaller network model was utilized to compare CPU and GPU implementations, indicating multi-factor speedup with the GPU version (Nair et al., 2015). The generalization may also be attributed to a subset of dense neurons being activated, weights adapted, and sparse recoding with a larger number of neurons in physiologically feasible geometries (Diwakar et al., 2011).

Motor articulation control with the cerebellum-inspired spiking neural network model

Cerebellum-inspired SNN could also predict trajectories by generating motor angles from end-effector coordinates without explicit kinematic representations. The network model represented the inverse kinematics transformations and predicted trajectories. For a single data point, the sensory consequence represented by the 3 end-effector coordinates (such as X, Y, and Z) was used to predict the 6 theta values, with the use of an encoder unit, 307 neurons, and a decoder for motor values, which converted to a "motor command" for the next point. For a short trajectory with four points (**Figure 7**), 1,228 neurons were needed for the robotic arm to trace the path. It was observed that the motor values changed more (**Figure 7A**) when compared with the end-effector coordinates, with negligible errors (**Figure 7B**). Except for the first motor value, all others were initially identical and had changed only in the third and fourth points, so the dotted line appeared to be superimposed (**Figure 7A**). The motor error and sensory error calculation were based on the difference between predicted and desired motor angle/end-effector coordinates. The motor error was used to update the weights at PF-PC and PC-DCN weights, and sensory error was used to update the weights at the MF-GrC. After training the network, a trajectory was plotted, and the change in features and output values showed

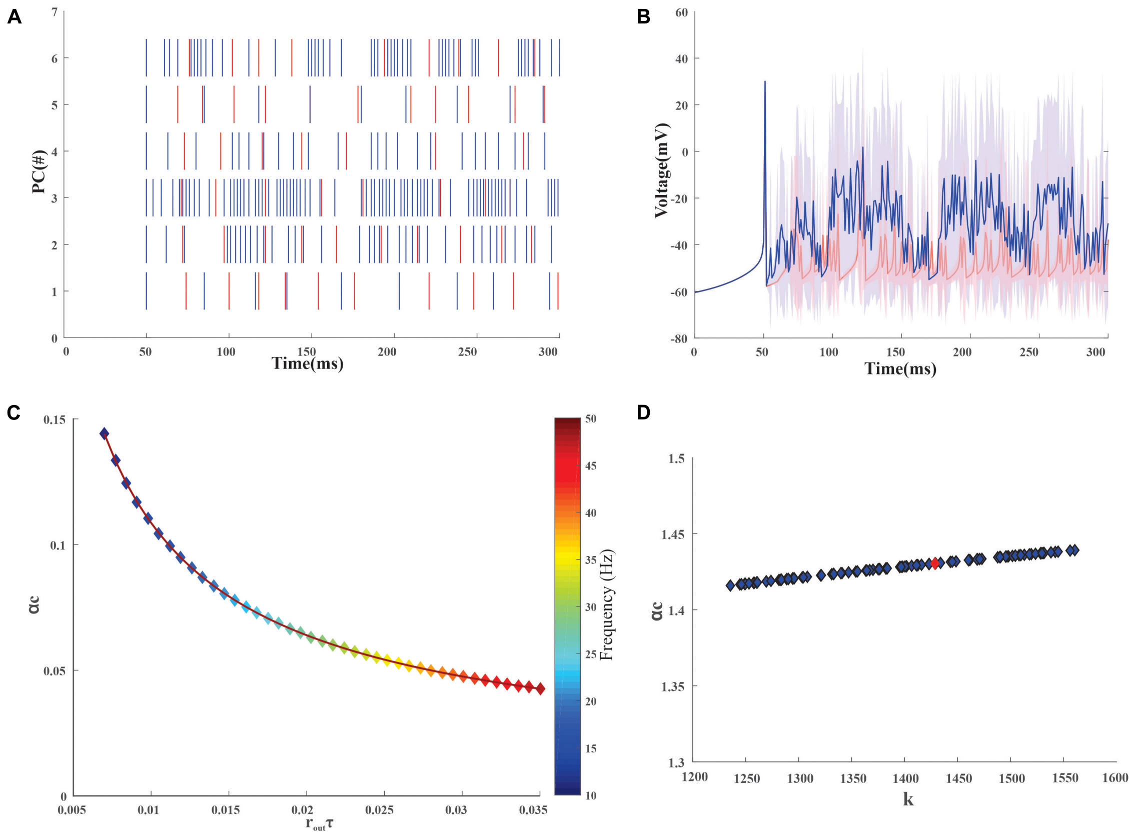

FIGURE 5
Firing dynamics and quantification of stored input-output association in the Purkinje layer neurons. **(A)** Spike variability of the 6 Purkinje neurons in the network, pre (red) and post (blue) learning. As learning adaptation improves, the firing frequency of the PC was also observed to increase. **(B)** Superimposition of the average firing dynamics of PC (pre and post) showed an increase in the firing frequency after learning adaptation with an increase in standard deviation **(C)**. Storage capacity (α_c) of PC with a firing frequency of 30–500 Hz was calculated and as mean firing output spike time ($r_{out}\tau$) increases, the storage capacity decreases gradually **(D)**. Storage capacity (α_c) of PC with stability constant (κ) representing the storage of new patterns at the synapse. Purkinje neuron (red dot) with a membrane time constant (τ_m) of 64 ± 17 ms was used for the reconstruction of the network.

minimal error suggesting the algorithm minimized both the sensory and motor error (see **Supplementary Figure 4**). With network configuration that matched the cerebellum's cytoarchitecture and connectivity, the outputs matched our previous implementation (Vijayan et al., 2017). The model mimicked the feedback controller functioning like the olivo-cerebellar circuit where the PC provided negative feedback to IO through the DCN, thereby reducing the network errors.

Discussion

The cerebellar spiking model was an abstraction of a bio-inspired architecture and could reconstruct the cerebellar network to predict and classify multimodal input. The model reduced multidimensional data using spatiotemporal characteristics of its spiking neurons and employed plasticity rules of the neuronal circuit. The model also suggests that the cerebellar microzone functions may have strong correlations to data encoding properties and may show adaptations while fine-tuning movement tasks. The current algorithm suggests granular layer geometries may help concurrent and implicit executions of large-scale pattern recoding problems, as shown by the model's feature classification.

Pattern classification depended on granule neuron encoding of the stimuli, purkinje cells temporal dynamics, and induced plasticity

MF inputs allow pattern representation through an abundance of the granule cells engaging the modalities of sensory patterns by similar encoding mechanisms. The network model used target representation to discriminate patterns from inputs. Multimodal conveyance of sensory signals through MF and the error signals by CF suggested by the Marr–Albus–Ito

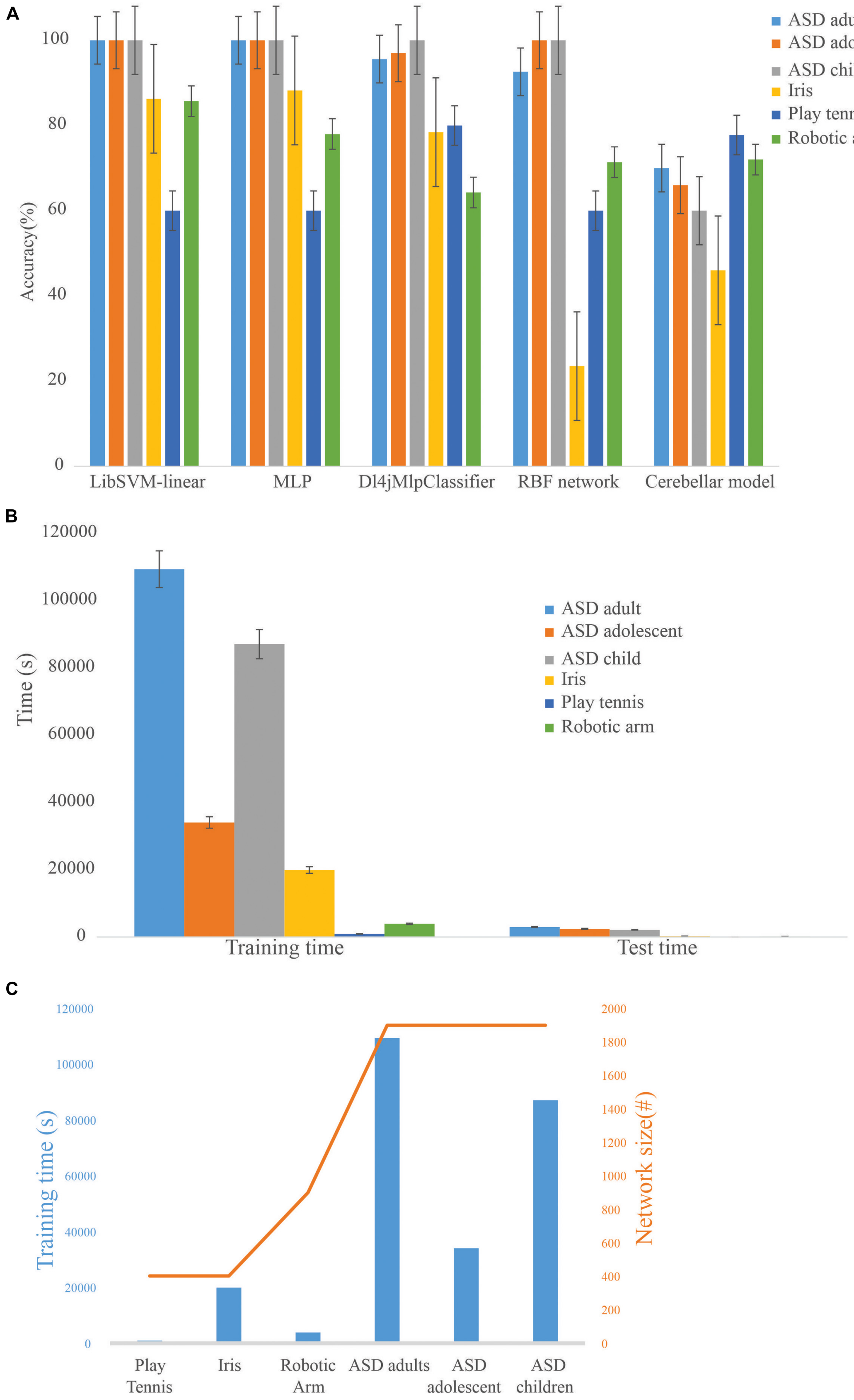

FIGURE 6

Comparison of the cerebellar model with machine learning algorithms on standard datasets **(A)**. Training accuracy of machine learning algorithms on different datasets showed that the cerebellar model could be used on a flexible dataset and rote learning was minimized. **(B)** Computational time required by the cerebellar model during the training and testing phase on various datasets. **(C)** Dependency of network size on training time.

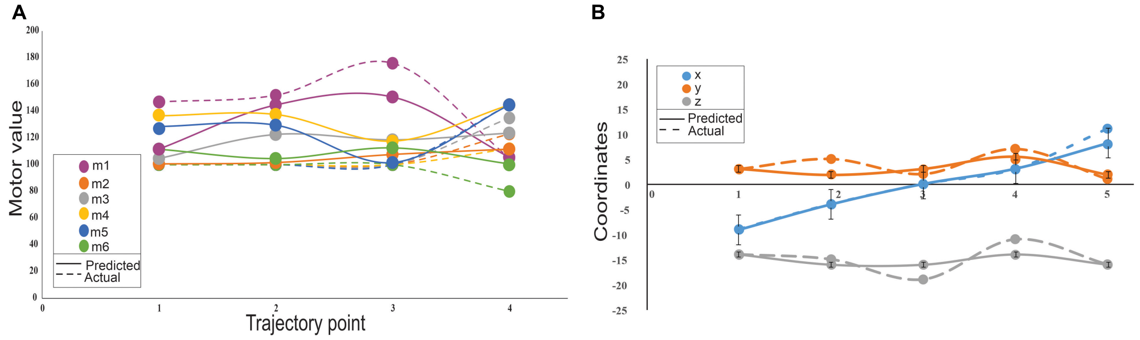

FIGURE 7
Variability on sensory and motor prediction for a 4-point trajectory by a low-cost arm. **(A)** Motor variability: Variability of the motor predictions was carried out using the cerebellar model that predicts the motor consequences from sensory commands. Motor errors were calculated from the actual values (dotted lines) and predicted (solid lines) for 6 motors. **(B)** Sensory variability: Sensory consequences of the predicted motor values were calculated using forward kinematics and errors associated were calculated as the difference from the actual to the predicted end-effector coordinates. Sensory error representations showed minimal variations and represented in the error bars some of which were negligibly small and very insignificant to be displayed.

theory may be an efficient representation for the cerebellum-inspired classifiers to discriminate specific patterns using the feed-forward and feedback mechanisms of the granular layer.

The algorithm employed time windowing to recode patterns, although in this current case study, only a small subset of different unique patterns was sufficient for the datasets used. In the cerebellum, GrC recoding uses the phenomena of time windowing (D'Angelo and Zeeuw, 2008), and the activation patterns in this one-to-many mapping parallel processing capabilities. The simulations suggest that time windows allow incorporating the parallel convolutional processing capability into the encoding performed by the granule cells, suggesting a specific convergence–divergence geometry could encode multiple pattern abstractions. A similar subset of GrCs may handle a different kind of task based on the same time window. GoC-GrC inhibition was crucial for pattern discrimination at the granule cell and supported the experimental data related to the time window hypothesis (Solinas et al., 2010). Inhibition-induced synchronicity could be relevant for pattern identification in the PC.

Reversible learning attributed to bidirectional plasticity at the MF-GrC and PF-PC synapses may be simplified as in the implemented model for recurrent learning tasks. A key role of such implementations would be a generalization task as reported in recent studies, and the current implementation employs the generalization to sparse and dense computations predicted by local field potential reconstructions (Diwakar et al., 2011). The model's generalization ability depended on the overlap or intersection of activated neurons in the granule layer. Like the learning capacity of a biological organism in learning from one experience to another, feature generalization looks at the stored neural responses and uses the same responses to mimic similar input patterns. A weight change is the main difference between connections. In this current study, the SI was used to measure the feature generalization capability at the granule layer as the input data were encoded to MFs. **Figure 4A** shows that some neurons respond to one type of input and some do not, but most neurons follow a normal distribution representing the generalization capability. The model network can be extended to a large number of clusters of GrC (randomness can be replaced with more biologically relevant jitter) to compare experimental patterns and their ensemble manifestations.

The cerebellar network model implicitly reconstructed inverse kinematics of a 6-degree of freedom robotic arm

A cluster of neurons may encode the position into a matching transformation representing the force position matrix. In the model, the kinematic transformation depends on the granular layer circuit and the consequential PC output by involving transformations of the network's spatiotemporal domain and space-force geometry. In the CISNN representation, a single joint was characterized by a collection of neurons attributing to the concept of different body muscles being controlled by different lobules in the cerebellar cortex (Bolk, 1906) and hence used the same to scale up the models based on the trajectory for better performance. By disabling certain GrCs during simulations, it was observed that errors in trajectory tracking increased, and the robotic arm could not complete the desired trajectory mapping. ANN-based single-layered architecture with a feed-forward granular layer (Bratby et al., 2016) or a feed-forward Purkinje layer (Clopath and Brunel, 2013) have been previously used for pattern separation or classification. Still, an SNN with multiple layers would be better for mathematically independent tasks. The somatotopic organization of the cerebellum at a functional level may involve similar representations by the clusters of granule neurons at the input layer, and modeling-based information processing relies on the connection geometries and the afferent pathways that bring in the information. The CISNN employs the coding aspects of the cerebellum's input layer and the discriminability of the Purkinje layer for performing sensory guidance of movement.

Quantitative measure of the memory representation

A subset of seven unique recoded MF patterns for each input feature contributed to the distributed representation of the granule neurons in the network. A random combination of the different stimuli causes the average granule layer responses to be uncorrelated and more identical or remain in a quiescent state. But as the inhibitory neurons are introduced to the granular layer, the sparseness increases, causing variability in the neuronal connections. Changes in weight and membrane time constant were some of the other factors that affect the sparseness, which was not considered in the study. Storage of random input-output associations in the Purkinje cell was estimated using storage capacity. Compared with the granule cells, the Purkinje neuron had a higher storage capacity which could be attributed to the bistable nature of the cell and its large dendritic arbor and that memory consolidation happened at the dendritic synapses. Studies (Li et al., 2022) had shown that higher information processing capacity led to lower neuronal activity and faster responses, suggesting that during the learning phase, neural engagement increased abruptly at the start of learning and then gradually declined. For GrCs, the storage capacity reduced steeply with firing frequency proposing the role of associative mapping over the storage of input-output associations and was associated with the Purkinje cells in real circuits. As the mean firing rate increased, the storage capacity decreased gradually, as observed in studies (Memmesheimer et al., 2014) denoting an inverse relation between r_{out} and storage capacity.

Cerebellar network model as a deep learning algorithm

Cerebellum input divergence and PF-PC mapping may serve as the convolutive coding layers employed by other DLN. A typical DLN involved multiple convolution layers, which may be the function of classification-related microcircuits in the cerebellum. In the CISNN, these convolution layers were represented by MF-GrC, MF-GoC-GrC, and GrC-PC transformations and their learning rules (Jörntell and Hansel, 2006) and match experimental data of the neurons performing temporal and combinatorial operations (Mapelli et al., 2010; Gilmer and Person, 2017). Taking advantage of these dynamics in ANN may help concurrent and implicit discriminations of large-scale patterns. From an algorithmic standpoint, the cerebellum-inspired deep learning model is scalable based on the input features and user-defined MFs and can accommodate multimodal input patterns functioning as a high dimensional coder for spatiotemporal data. Projected spatiotemporal coding properties of the cerebellum allowed converting real-world data into spike trains and were used to classify the dataset. Comparing the cerebellar model with different datasets and other standard ML algorithms suggests that the model functions as a good classifier with considerable accuracy, which tends to increase with epoch when the learning rate was reduced, suggesting an efficient learning mechanism. Even though the performance of other well-known algorithms, such as MLP, Dl4jMlpClassifier, and RBF networks fractionally outperform the cerebellar model in terms of accuracy, the model was sufficient enough to do the task of classification as well as trajectory prediction for a low-cost sensor-free robotic articulator. The robotic arm was developed in the lab whose total cost was < 5$ (Vijayan et al., 2013), and a network model with fewer neurons was able to perform the specified tasks. The cerebellum-inspired spiking neural network was optimized in less than 50 epochs suggesting that employing a feed-forward cerebellum-based model could be used as a supervised classifier and predictor for big data.

Scaling up the spiking network model and modeling cerebellar function

MF-GrC-GoC is an internal loop that monitors the input by providing back-and forth-signals at the GrC-GoC connection. Recurrent inhibitory loops play a significant role in error minimization in the cerebellum. Some loops are computationally costly and were avoided. This may need to be reconsidered for big data operations. For computational efficacy, the algorithm did not model silent or non-spiking granule cells and the optimization of learning was based directly on the abstractions of the non-silent cells. The burden on learning was optimized by the granule cell's silent/non-silent ratio, which could be redefined using convergence–divergence.

The coordination of articulator joints may be understood from the spiking reconstruction as in CISNN, and the scalability of the cerebellar input layer clusters can be exploited to understand the cerebellar functions in the context of robotics computationally. The algorithm does not include any sensor or visual-based inputs, thereby decoupling non-motor sensory inputs from the cerebellum-based internal models. As a deep learning algorithm, the CISNN poses many challenges, such as overfitting a model to its training data.

Prediction accuracy relied on the role of dual representation of errors (such as sensory error and motor error) (Popa et al., 2012), which may be approximated in the learning process as the model generalized the prediction process. The model's inability to discriminate sensory or motor errors could be attributed to the lack of CF-related error feedback in the hardware we employed to generate the trajectories. Even after the training phase, some of the weights remained unoptimized due to the weight assignments, which can be resolved by introducing heuristic initialization, such as Xavier's or Kaiming's initialization that employed ReLu, leaky ReLu, or tanh functions.

Optimizing learning rates for the different neurons was a big challenge, requiring tweaking the different parameters without changing the neuronal dynamics. Larger models yield better performance when compared with smaller models (Li et al., 2013). For scaling up the model, a larger dataset or more MFs have to be introduced, which in turn creates a need for high computation power for running the simulations. Scaled-up SNN are now used to learn spatiotemporal dynamic patterns of biosignals and map neuronal connections created during the learning process (Gholami Doborjeh et al., 2018).

Like DLN models, the CISNN architecture may require scaling optimizations, reducing the time complexity for large datasets (with the four Vs) to solve real-world problems. Scaling the model for very large datasets required parallelization of the input layer geometry, and faster learning in terms of time complexity may be attained using GPU-based processing units. The advantage of this framework is that the algorithm can be reimplemented on GPGPUs and FPGA hardware. Although we tested with smaller sizes and resolutions, these parallel GPU implementations may allow scaling this cerebellum-inspired spiking neural network for big data and streaming data classifications.

The model may be relevant to reconstruct neurological disorders mimicking programmed failure of robotic motor joints during trajectory mapping by articulators. Although we did not implement all the 16 known forms of plasticity (Gao et al., 2012; D'Angelo et al., 2016a) and reinforcement learning in the MF-GrC pathway (Wagner et al., 2017), this cerebellum-inspired implementation currently serves as a reductionist mapping between the convergence–divergence ratios of neurons and connection geometries and could act as a testbed for understanding dysfunctions. The model is relevant and can be extended for the movement-related reconstruction of inverse kinematics as well as modeling coordination-related changes.

Comparison with other cerebellar models

The current model was reconstructed based on known microcircuitry of the cerebellum (Eccles, 1967), electrophysiological behavior (D'Angelo et al., 2001), and significant plasticity rules (Mapelli et al., 2015). Even though there are several models available that look at the different aspects of the cerebellum (Casellato et al., 2012; Garrido et al., 2013; Antonietti et al., 2015; D'Angelo et al., 2016b; Luque et al., 2016; Yamaura et al., 2020; Kuriyama et al., 2021), the present model covers certain aspects of the cerebellum while some loops are skipped. Initial models looked at only single-layered neurons (Marr, 1969; Albus, 1971) which were extended with many other layers and plasticity rules. Models such as EDLUT (Carrillo et al., 2008) are more focused on the event-driven simulation scheme based on lookup tables that would reduce the time involved with the numerical calculation. In the current model, numerical calculations are introduced to best capture the processed information and filter the information that has to be sent. For specific tasks, there are networks with cells arranged in 3D structure geometry that can be simulated with supercomputers giving it the dynamics of the cerebellum (Yamaura et al., 2020). Compared with one of the closest earlier models (Hausknecht et al., 2016) where multiple modular cerebellar circuits have been considered to train different tasks, here the current model has considered a single module of the cerebellar architecture which could reduce the complexity of the network.

The cerebellar model was made scalable based on input data with fewer cells and loops, which has a crucial role in predicting and classifying learning data. The uniqueness of the current model is the incorporation of the recoding and associative mapping properties of the cerebellar granular layer with excitatory GrC and an inhibitory GoC, pattern discrimination property at the Purkinje layer, and the interpretational application of DCN along with an encoding and decoding module which involves the transformation of real-world data to spiking information and vice versa.

Conclusion

The functions of the cerebellum in terms of fast computations and self-organizing the mapping of input patterns are yet to be better understood. This algorithmic model is a computational step in interpreting the complex control behind pattern reorganization of the motor and other signals while exploring the cerebellar architecture as a deep learning model. While error analysis, data sparseness, optimal adaptation of weights, and the relationships to known plasticity rules remain to be implemented, this model allows generalization within the context of spiking neural architecture-based pattern classification.

This cerebellar network model would allow to explore multimodal circuit-dependent pattern discrimination and sensorimotor transformations. The proposed algorithm shows potential in exhibiting dynamical properties, such as (1) improved accuracy and prediction with learning, (2) autonomous update, (3) neuronal network scalability with distributed plasticity. Given the theoretical goals of such an implementation, this model will also be adapted and extended to include recurrent loops and adaptive reinforcement learning and can be employed on big datasets. The model can be implemented on VLSI hardware and FPGA boards because of the simpler and modular mathematical operations.

The proposed model could help understand movement-related pattern recognition, such as playing a game of chess and improve models for motor articulation control by storing as well as predicting trajectories for path optimization, such as moving a cup of water. It could be used as a prediction mechanism for pathophysiological conditions and aid in BCI and neuroprosthesis like an artificial limb can be used as a deep learning model to understand the cerebellar function and dysfunctions.

With the cerebellar information and cellular losses that occur in neurological conditions, these models can be used to fine-tune tasks involving various mathematical complexity without significantly adapting the connectivity and circuit. With that, we believe this model may have prominent usability roles for experimental neuroscientists to hypothesize and explore other functions.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found here: https://github.com/compneuro/DLCISNN.

Author contributions

AV and SD conceived and developed the method, analyzed the data, and wrote the manuscript. Both authors read and approved the final manuscript.

Funding

This study was partially supported by the Department of Science and Technology (grant DST/CSRI/2017/31), from the Government of India, Amrita Vishwa Vidyapeetham, and embracing the world research for a cause initiative.

Acknowledgments

This project derives direction and ideas from the Chancellor of Amrita Vishwa Vidyapeetham, Mātā Amritānandamayī Devī. We would like to thank Dhanush Kumar, Joshi Alphonse, Anandhu Presannan, Nutakki Chaitanya, Sreedev Radhakrishnan, and Manjusha Nair at Amrita Vishwa Vidyapeetham for their assistance related to this manuscript.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fnins.2022.909146/full#supplementary-material

References

Albus, J. S. (1971). A theory of cerebellar function. *Math. Biosci.* 10, 25–61. doi: 10.1016/0025-5564(71)90051-4

Albus, J. S. (1975). A new approach to manipulator control: The cerebellar model articulation controller(CMAC). *J. Dyn. Syst. Meas. Control.* 97, 220–227. doi: 10.1115/1.3426922

Angermueller, C., Pärnamaa, T., Parts, L., and Stegle, O. (2016). Deep learning for computational biology. *Mol. Syst. Biol.* 12:878. doi: 10.15252/msb.20156651

Antonietti, A., Casellato, C., Garrido, J., Luque, N. R., Naveros, F., Ros, E., et al. (2015). Spiking neural network with distributed plasticity reproduces cerebellar learning in eye blink conditioning paradigms. *IEEE Trans. Biomed. Eng.* 63:210-219. doi: 10.1109/TBME.2015.2485301

Babadi, B., and Sompolinsky, H. (2014). Sparseness and expansion in sensory representations. *Neuron* 83, 1213–1226. doi: 10.1016/j.neuron.2014.07.035

Baladron, J., and Hamker, F. H. (2015). A spiking neural network based on the basal ganglia functional anatomy. *Neural Netw.* 67, 1–13. doi: 10.1016/j.neunet.2015.03.002

Billings, G., Piasini, E., Lorincz, A., Nusser, Z., and Silver, R. A. (2014). Network structure within the cerebellar input layer enables lossless sparse encoding. *Neuron* 83, 960–974. doi: 10.1016/j.neuron.2014.07.020

Bolk, L. (1906). *Das cerebellum der saugetiere: Eine vergleichend anatomische untersuchung*. Jena: Fischer.

Bratby, P., Sneyd, J., and Montgomery, J. (2016). Computational architecture of the granular layer of cerebellum-like structures. *Cerebellum* doi: 10.1007/s12311-016-0759-z

Brunel, N., Hakim, V., Isope, P., Nadal, J. P., and Barbour, B. (2004). Optimal information storage and the distribution of synaptic weights: Perceptron versus Purkinje cell. *Neuron* 43, 745–757. doi: 10.1016/S0896-6273(04)00528-8

Cajal, S. R. y (1911). *Histologie du sistéme nerveux de l'homme et des vertebras . Transl, L.* Azoulay Paris: Maloine.

Carrillo, R. R., Ros, E., Boucheny, C., and Coenen, O. J.-M. D. (2008). A real-time spiking cerebellum model for learning robot control. *Biosystems* 94, 18–27. doi: 10.1016/j.biosystems.2008.05.008

Casellato, C., Antonietti, A., Garrido, J. a, Carrillo, R. R., Luque, N. R., Ros, E., et al. (2014). Adaptive robotic control driven by a versatile spiking cerebellar network. *PLoS One* 9:e112265. doi: 10.1371/journal.pone.0112265

Casellato, C., Pedrocchi, A., Garrido, J. A., Luque, N. R., Ferrigno, G., D'Angelo, E., et al. (2012). "An integrated motor control loop of a human-like robotic arm: Feedforward, feedback and cerebellum-based learning," in *Proceedings of the IEEE RAS and EMBS international conference on biomedical robotics and biomechatronics*, 562–567. Rome. doi: 10.1109/BioRob.2012.6290791

Cathala, L., Brickley, S., Cull-Candy, S., and Farrant, M. (2003). Maturation of EPSCs and intrinsic membrane properties enhances precision at a cerebellar synapse. *J. Neurosci.* 23, 6074–6085. doi: 10.1523/JNEUROSCI.23-14-06074.2003

Clopath, C., and Brunel, N. (2013). Optimal properties of analog perceptrons with excitatory weights. *PLoS Computational Biol.* 9:e1002919. doi: 10.1371/journal.pcbi.1002919

Clopath, C., Nadal, J. P., and Brunel, N. (2012). Storage of correlated patterns in standard and bistable Purkinje cell models. *PLoS Computational Biol.* 8:e1002448. doi: 10.1371/journal.pcbi.1002448

D'Angelo, E., Mapelli, L., Casellato, C., Garrido, J. A., Luque, N., Monaco, J., et al. (2016b). Distributed circuit plasticity: New clues for the cerebellar mechanisms of learning. *Cerebellum* 15, 139–151. doi: 10.1007/s12311-015-0711-7

D'Angelo, E., Antonietti, A., Casali, S., Casellato, C., Garrido, J. A., Luque, N. R., et al. (2016a). Modeling the cerebellar microcircuit: New strategies for a long-standing issue. *Front. Cell. Neurosci.* 10:176. doi: 10.3389/fncel.2016.00176

D'Angelo, E., Nieus, T., Maffei, A., Armano, S., Rossi, P., Taglietti, V., et al. (2001). Theta-frequency bursting and resonance in cerebellar granule cells: Experimental evidence and modeling of a slow k+-dependent mechanism. *J. Neurosci.* 21, 759–770. doi: 10.1523/JNEUROSCI.21-03-00759.2001

D'Angelo, E., Solinas, S., Mapelli, J., Gandolfi, D., Mapelli, L., and Prestori, F. (2013). The cerebellar golgi cell and spatiotemporal organization of granular layer activity. *Front. Neural Circ.* 7:93. doi: 10.3389/fncir.2013.00093

D'Angelo, E., Zeeuw, C. I. (2008). Timing and plasticity in the cerebellum: Focus on the granular layer. *Trends Neurosci.* 32, 30–40. doi: 10.1016/j.tins.2008.09.007

Dean, P., Porrill, J., Ekerot, C.-F., and Jorntell, H. (2010). The cerebellar microcircuit as an adaptive filter: Experimental and computational evidence. *Nat. Rev. Neurosci.* 11, 30–43. doi: 10.1038/nrn2756

Delvendahl, I., Straub, I., and Hallermann, S. (2015). Dendritic patch-clamp recordings from cerebellar granule cells demonstrate electrotonic compactness. *Front. Cell. Neurosci.* 9, 1–8. doi: 10.3389/fncel.2015.00093

Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S., and Pfeiffer, M. (2015). "Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing," in *Proceedings of the 2015 International Joint Conference on Neural Networks (IJCNN)*, Killarney. doi: 10.1109/IJCNN.2015.7280696

Diwakar, S., Lombardo, P., Solinas, S., Naldi, G., and D'Angelo, E. (2011). Local field potential modeling predicts dense activation in cerebellar granule cells clusters under LTP and LTD control. *PLoS One* 6:e21928. doi: 10.1371/journal.pone.0021928

Doya, K. (1999). What are the computations of the cerebellum, the basal ganglia and the cerebral cortex? *Neural Netw.* 12, 961–974. doi: 10.1016/S0893-6080(99)00046-5

Doya, K. (2000). Complementary roles of basal ganglia and cerebellum in learning and motor control. *Curr. Opin. Neurobiol.* 10, 732–739. doi: 10.1016/S0959-4388(00)00153-7

Eccles, J. C. (1967). Circuits in the cerebellar control of movement. *Proc. Natl. Acad. Sci. USA*. 58, 336–343. doi: 10.1073/pnas.58.1.336

Eccles, J. C. (1973). The cerebellum as a computer: Patterns in space and time. *J. Physiol.* 229, 1–32. doi: 10.1113/jphysiol.1973.sp010123

Eccles, J. C., Ito, M., and Szentagothal, J. (1967b). Internal workings of the brain. *Science* 158, 1439–1440. doi: 10.1126/science.158.3807.1439

Eccles, J. C., Ito, M., and Szentágothai, J. (1967a). *The cerebellum as a neuronal machine*. Berlin: Springer. doi: 10.1016/0013-4694(69)90099-6

Fiete, I. R., Hahnloser, R. H. R., Fee, M. S., and Seung, H. S. (2004). Temporal sparseness of the premotor drive is important for rapid learning in a neural network model of birdsong. *J. Neurophysiol.* 92, 2274–2282. doi: 10.1152/jn.01133.2003

Fontana, A. (2017). A deep learning-inspired model of the hippocampus as storage device of the brain extended dataset. *arXiv* [Preprint].

Fu, S. Y., Yang, G. S., and Kuai, X. K. (2012). A spiking neural network based cortex-like mechanism and application to facial expression recognition. *Comput. Intell. Neurosci.* 2012:946589 doi: 10.1155/2012/946589

Fujita, M. (1982). Adaptive filter model of the cerebellum. *Biol. Cybern.* 45, 195–206. doi: 10.1007/BF00336192

Gabbiani, F., Midtgaard, J., and Knopfel, T. (1994). Synaptic integration in a model of cerebellar granule cells. *J. Neurophysiol.* 72, 999–1009. doi: 10.1152/jn.1994.72.2.999

Galliano, E., Gao, Z., Schonewille, M., Todorov, B., Simons, E., Pop, A. S., et al. (2013). Silencing the majority of cerebellar granule cells uncovers their essential role in motor learning and consolidation. *Cell Rep.* 3, 1239–1251. doi: 10.1016/j.celrep.2013.03.023

Gao, Z., Beugen, B. J., van, Zeeuw, C. I., and De. (2012). Distributed synergistic plasticity and cerebellar learning. *Nat. Rev. Neurosci.* 13, 619–635. doi: 10.1038/nrn3312

Garrido, J. A., Luque, N. R., D'Angelo, E., and Ros, E. (2013). Distributed cerebellar plasticity implements adaptable gain control in a manipulation task: A closed-loop robotic simulation. *Front. Neural Circ.* 7:159. doi: 10.3389/fncir.2013.00159

Gholami Doborjeh, Z., Kasabov, N., Gholami Doborjeh, M., and Sumich, A. (2018). Modelling peri-perceptual brain processes in a deep learning spiking neural network architecture. *Sci. Rep.* 8:8912. doi: 10.1038/s41598-018-27169-8

Ghosh-Dastidar, S., and Adeli, H. (2009). A new supervised learning algorithm for multiple spiking neural networks with application in epilepsy and seizure detection. *Neural Netw.* 22, 1419–1431. doi: 10.1016/j.neunet.2009.04.003

Gilmer, J. I., and Person, A. L. (2017). Morphological constraints on cerebellar granule cell combinatorial diversity. *J. Neurosci.* 37, 12153–12166. doi: 10.1523/JNEUROSCI.0588-17.2017

Girard, B., Lienard, J., Gutierrez, C. E., Delord, B., and Doya, K. (2020). A biologically constrained spiking neural network model of the primate basal ganglia with overlapping pathways exhibits action selection. *Eur. J. Neurosci.* 53, 2254–2277. doi: 10.1111/ejn.14869

Haith, A., and Vijayakumar, S. (2009). Implications of different classes of sensorimotor disturbance for cerebellar-based motor learning models. *Biol. Cybern.* 100, 81–95. doi: 10.1007/s00422-008-0266-5

Hajj, N., and Awad, M. (2018). "A biologically inspired deep neural network of basal ganglia switching in working memory tasks," in *Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence, SSCI 2017 - Proceedings* (Institute of Electrical and Electronics Engineers Inc.),Honolulu, HI. doi: 10.1109/SSCI.2017.8285364

Hausknecht, M., Li, W. K., Mauk, M., and Stone, P. (2016). Machine learning capabilities of a simulated cerebellum. *IEEE Trans. Neural Netw. Learn. Syst.* 28:510-522. doi: 10.1109/TNNLS.2015.2512838

Herzfeld, D. J., Hall, N. J., Tringides, M., and Lisberger, S. G. (2020). Principles of operation of a cerebellar learning circuit. *Elife* 9:e55217. doi: 10.7554/eLife.55217

Inagaki, K., and Hirata, Y. (2017). Computational theory underlying acute vestibulo-ocular reflex motor learning with cerebellar long-term depression and long-term potentiation. *Cerebellum* 16, 827–839. doi: 10.1007/s12311-017-0857-6

Ito, M. (1972). Neural design of the cerebellar motor control system. *Brain Res.* 40, 81–84. doi: 10.1016/0006-8993(72)90110-2

Ito, M. (1982). Cerebellar control of the vestibulo-ocular reflex- around the flocculus hypothesis. *Annu. Rev. Neurosci* 5, 275–296. doi: 10.1146/annurev.ne.05.030182.001423

Ito, M. (2002). Historical review of the significance of the cerebellum and the role of purkinje cells in motor learning. *Ann. N Y. Acad. Sci.* 978, 273–288. doi: 10.1111/j.1749-6632.2002.tb07574.x

Ito, M. (2006). Cerebellar circuitry as a neuronal machine. *Prog. Neurobiol.* 78, 272–303. doi: 10.1016/j.pneurobio.2006.02.006

Jörntell, H., and Hansel, C. (2006). Synaptic memories upside down: Bidirectional plasticity at cerebellar parallel fiber-Purkinje cell synapses. *Neuron* 52, 227–238. doi: 10.1016/j.neuron.2006.09.032

Kawato, M. (1999). Internal models for motor control and trajectory planning. *Curr. Opin. Neurobiol.* 9, 718–727. doi: 10.1016/S0959-4388(99)00028-8

Kawato, M., and Gomi, H. (1992). The cerebellum and VOR/OKR learning models. *Trends Neurosci.* 15, 445–453. doi: 10.1016/0166-2236(92)90008-V

Kawato, M., Kuroda, S., and Schweighofer, N. (2011). Cerebellar supervised learning revisited: Biophysical modeling and degrees-of-freedom control. *Curr. Opin. Neurobiol.* 21, 791–800. doi: 10.1016/j.conb.2011.05.014

Kawato, M., Ohmae, S., Hoang, H., and Sanger, T. (2020). 50 years since the Marr, ito, and albus models of the cerebellum. *Neuroscience*. 62, 151-174. . doi: 10.1016/j.neuroscience.2020.06.019

Kemp, K. C., Cook, A. J., Redondo, J., Kurian, K. M., Scolding, N. J., and Wilkins, A. (2016). Purkinje cell injury, structural plasticity and fusion in patients with Friedreich's ataxia. *Acta Neuropathol. Commun.* 4:53. doi: 10.1186/s40478-016-0326-3

Kettner, R. E., Mahamud, S., Leung, H. C., Sitkoff, N., Houk, J. C., Peterson, B. W., et al. (1997). Prediction of complex two-dimensional trajectories by a cerebellar model of smooth pursuit eye movement. *J. Neurophysiol.* 77, 2115–2130. doi: 10.1049/iet-cta.2010.0464

Khaliq, Z. M., and Raman, I. M. (2005). Axonal propagation of simple and complex spikes in cerebellar Purkinje neurons. *J. Neurosci.* 25, 454–463. doi: 10.1523/JNEUROSCI.3045-04.2005

Khan, M. M., Lester, D. R., Plana, L. A., Rast, A., Jin, X., Painkras, E., et al. (2008). "SpiNNaker: Mapping neural networks onto a massively-parallel chip multiprocessor," in *Proceddings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)* (IEEE), 2849–2856. Hong Kong doi: 10.1109/IJCNN.2008.4634199

Kindel, W. F., Christensen, E. D., and Zylberberg, J. (2019). Using deep learning to probe the neural code for images in primary visual cortex. *Journal of Vision* 19, 1–12. doi: 10.1167/19.4.29

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biol. Cybern.* 43, 59–69. doi: 10.1007/BF00337288

Kuriyama, R., Casellato, C., D'Angelo, E., and Yamazaki, T. (2021). Real-time simulation of a cerebellar scaffold model on graphics processing units. *Front. Cell. Neurosci.* 15:623552. doi: 10.3389/fncel.2021.623552

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2017). Building machines that learn and think like people. *Behav. Brain Sci.* 40:e253. doi: 10.1017/S0140525X16001837

Li, T., Zheng, Y., Wang, Z., Zhu, D. C., Ren, J., Liu, T. S., et al. (2022). Brain information processing capacity modeling. *Sci. Rep.* 12, 2174. doi: 10.1038/s41598-022-05870-z

Li, W.-K., Hausknecht, M. J., Stone, P. H., and Mauk, M. D. (2013). Using a million cell simulation of the cerebellum: Network scaling and task generality. *Neural Netw.* 47, 95–102. doi: 10.1016/j.neunet.2012.11.005

Lichman, M. (2013). "UCI Machine Learning Repository," in *University of California, Irvine, School of Information and Computer Sciences*. California: Donald Bren School of Information and Computer Sciences.

Luo, J., Coapes, G., Mak, T., Yamazaki, T., Tin, C., and Degenaar, P. (2016). Real-time simulation of passage-of-time encoding in cerebellum using a scalable FPGA-based system. *IEEE Trans. Biomed. Circ. Syst.* 10, 742–753. doi: 10.1109/TBCAS.2015.2460232

Luo, X., Qu, H., Zhang, Y., and Chen, Y. (2019). First error-based supervised learning algorithm for spiking neural networks. *Front. Neurosci.* 13:559. doi: 10.3389/fnins.2019.00559

Luque, N. R., Garrido, J. A., Carrillo, R. R., Coenen, O. J.-M. D., and Ros, E. (2011). Cerebellar input configuration toward object model abstraction in manipulation tasks. *IEEE Trans. Neural Netw.* 22, 1321–1328. doi: 10.1109/TNN.2011.2156809

Luque, N. R., Garrido, J. A., Naveros, F., Carrillo, R. R., D'Angelo, E., and Ros, E. (2016). Distributed cerebellar motor learning: A spike-timing-dependent plasticity model. *Front. Comput. Neurosci.* 10, 1–22. doi: 10.3389/fncom.2016.00017

Maex, R., and De Schutter, E. (1998). Synchronization of golgi and granule cell firing in a detailed network model of the cerebellar granule cell layer. *J. Neurophysiol.* 80, 2521–2537. doi: 10.1152/jn.1998.80.5.2521

Manto, M., Bower, J. M., Conforto, A. B., Delgado-García, J. M., Da Guarda, S. N. F., Gerwig, M., et al. (2012). Consensus paper: Roles of the cerebellum in motor control-the diversity of ideas on cerebellar involvement in movement. *Cerebellum* 11, 457–487. doi: 10.1007/s12311-011-0331-9

Mapelli, J., Gandolfi, D., and D'Angelo, E. (2010). Combinatorial responses controlled by synaptic inhibition in the cerebellum granular layer. *J. Neurophysiol.* 103, 250–261. doi: 10.1152/jn.00642.2009

Mapelli, L., Pagani, M., Garrido, J. A., and D'Angelo, E. (2015). Integrated plasticity at inhibitory and excitatory synapses in the cerebellar circuit. *Front. Cell. Neurosci.* 9:169. doi: 10.3389/fncel.2015.00169

Marr, D. (1969). A theory of cerebellar cortex. *J. Physiol.* 202, 437–470. doi: 10.1113/jphysiol.1969.sp008820

Masoli, S., Solinas, S., and D'Angelo, E. (2015). Action potential processing in a detailed Purkinje cell model reveals a critical role for axonal compartmentalization. *Front. Cell. Neurosci.* 9:47. doi: 10.3389/fncel.2015.00047

Mauk, M. D., Steinmetz, J. E., and Thompson, R. F. (1986). Classical conditioning using stimulation of the inferior olive as the unconditioned stimulus. *Proc. Natl. Acad. Sci. U.S.A.* 83, 5349–5353. doi: 10.1073/pnas.83.14.5349

Medina, J. F., and Mauk, M. D. (2000). Computer simulation of cerebellar information processing. *Nat. Neurosci.* 3, 1205–1211. doi: 10.1038/81486

Medini, C., Nair, B., D'Angelo, E., Naldi, G., and Diwakar, S. (2012). Modeling spike-train processing in the cerebellum granular layer and changes in plasticity reveal single neuron effects in neural ensembles. *Comput. Intell. Neurosci.* 2012:17. doi: 10.1155/2012/359529

Medini, C., Vijayan, A., D'Angelo, E., Nair, B., and Diwakar, S. (2014). "Computationally Efficient Bio-realistic Reconstructions of Cerebellar Neuron Spiking Patterns," in *Proceedings of the Interdisciplinary Advances in Applied Computing - ICONIAAC '14,Proceedings of the 2014 International Conference on*, 1–6. Sydney. doi: 10.1145/2660859.2660961

Memmesheimer, R. M., Rubin, R., Ölveczky, B. P., and Sompolinsky, H. (2014). Learning precisely timed spikes. *Neuron* 82, 1–14. doi: 10.1016/j.neuron.2014.03.026

Nair, M., Nair, B., and Diwakar, S. (2014). "Large-Scale Simulations of Cerebellar Microcircuit Relays using Spiking Neuron on GPUs," in *Eleventh International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, Cambridge.

Nair, M., Nair, B., and Diwakar, S. (2015). "GPGPU Implementation of a Spiking Neuronal Circuit Performing Sparse Recoding," in *Computational Intelligence Methods for Bioinformatics and Biostatistics* (Springer International Publishing), Cambridge, UK 285–297. doi: 10.1007/978-3-319-24462-4_24

Naud, R., Marcille, N., Clopath, C., and Gerstner, W. (2008). Firing patterns in the adaptive exponential integrate-and-fire model. *Biol. Cybern.* 99, 335–347. doi: 10.1007/s00422-008-0264-7

Naveros, F., Garrido, J. A., Carrillo, R. R., Ros, E., and Luque, N. R. (2017). Event- and time-driven techniques using parallel CPU-GPU Co-processing for spiking neural networks. *Front. Neuroinformatics* 11:7. doi: 10.3389/fninf.2017.00007

Nieus, T., Sola, E., Mapelli, J., Saftenku, E., Rossi, P., and D'Angelo, E. (2006). LTP regulates burst initiation and frequency at mossy fiber-granule cell synapses of rat cerebellum: Experimental observations and theoretical predictions. *J. Neurophysiol.* 95, 686–699. doi: 10.1152/jn.00696.2005

Nowak, D. a, Topka, H., Timmann, D., Boecker, H., and Hermsdörfer, J. (2007). The role of the cerebellum for predictive control of grasping. *Cerebellum* 6, 7–17. doi: 10.1080/14734220600776379

Nutakki, C., Vijayan, A., Sasidharakurup, H., Nair, B., Achuthan, K., and Diwakar, S. (2016). "Low-Cost Robotic Articulator as an Online Education tool: Design, Deployment and Usage," in *Proceedings of IEEE International Conference on Robotics and Automation for Humanitarian Applications* (Kollam). doi: 10.1109/RAHA.2016.7931888

Ogawa, T., and Kanada, H. (2010). Solution for Ill-posed inverse kinematics of robot arm by network inversion. *J. Robot.* 2010:5720163. doi: 10.1155/2010/870923

Passot, J. B., Luque, N. R., and Arleo, A. (2010). "Internal models in the cerebellum: A coupling scheme for online and offline learning in procedural tasks," in *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* 6226 LNAI, Berlin. 435–446. doi: 10.1007/978-3-642-15193-4_41

Pehlevan, C., and Sompolinsky, H. (2014). Selectivity and sparseness in randomly connected balanced networks. *PLoS One* 9:e89992. doi: 10.1371/journal.pone.0089992

Pfeiffer, M., and Pfeil, T. (2018). Deep learning with spiking neurons: Opportunities and challenges. *Front. Neurosci.* 12:774. doi: 10.3389/fnins.2018.00774

Pinzon-Morales, R.-D., and Hirata, Y. (2013). "Cerebellar-inspired bi-hemispheric neural network for adaptive control of an unstable robot," in *Proceedings of the 2013 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC)*, Rio de Janeiro. doi: 10.1109/BRC.2013.6487536

Popa, L. S., Hewitt, a. L, and Ebner, T. J. (2012). Predictive and feedback performance errors are signaled in the simple spike discharge of individual purkinje cells. *J. Neurosci.* 32, 15345–15358. doi: 10.1523/JNEUROSCI.2151-12.2012

Popa, L. S., Streng, M. L., Hewitt, A. L., and Ebner, T. J. (2016). The errors of our ways: Understanding error representations in cerebellar-dependent motor learning. *Cerebellum* 15, 93–103. doi: 10.1007/s12311-015-0685-5

Porrill, J., Dean, P., and Stone, J. V. (2004). Recurrent cerebellar architecture solves the motor-error problem. *Proc. Biol. Sci.* 271, 789–796. doi: 10.1098/rspb.2003.2658

Redondo, J., Kemp, K., Hares, K., Rice, C., Scolding, N., and Wilkins, A. (2015). Purkinje cell pathology and loss in multiple sclerosis cerebellum. *Brain Pathol.* 25, 692–700. doi: 10.1111/bpa.12230

Rodriguez-Fornells, A., Kurzbuch, A. R., and Münte, T. F. (2002). Time course of error detection and correction in humans: Neurophysiological evidence. *J. Neurosci.* 22, 9990–9996. 22/22/9990 [pii] doi: 10.1523/JNEUROSCI.22-22-09990.2002

Ros, E., Carrillo, R., Ortigosa, E. M., Barbour, B., and Agís, R. (2006). Event-driven simulation scheme for spiking neural networks using lookup tables to characterize neuronal dynamics. *Neural Comput.* 18, 2959–2993. doi: 10.1162/neco.2006.18.12.2959

Rössert, C., Dean, P., and Porrill, J. (2015). At the edge of chaos: How cerebellar granular layer network dynamics can provide the basis for temporal filters. *PLoS Comput. Biol.* 11, 1–28. doi: 10.1371/journal.pcbi.1004515

Roth, A., and Hausser, M. (2001). Compartmental models of rat cerebellar Purkinje cells based on simultaneous somatic and dendritic patch-clamp recordings. *J. Physiol.* 535, 445–472. doi: 10.1111/j.1469-7793.2001.00445.x

Rubin, R., Monasson, R., and Sompolinsky, H. (2010). Theory of spike timing based neural classifiers. *Phys. Rev. Lett.* 105:218102. doi: 10.1103/PhysRevLett.105.218102

Sausbier, M., Hu, H., Arntz, C., Feil, S., Kamm, S., Adelsberger, H., et al. (2004). Cerebellar ataxia and Purkinje cell dysfunction caused by Ca2+-activated K+ channel deficiency. *Proc. Natl. Acad. Sci. U.S.A.* 101, 9474–9478. doi: 10.1073/pnas.0401702101

Schweighofer, N., and Arbib, M. A. (1998). A model of cerebellar metaplasticity. *Learn. Mem.* 4, 421–428. doi: 10.1101/lm.4.5.421

Schweighofer, N., Arbib, M. A., and Dominey, P. F. (1996). A model of the cerebellum in adaptive control of saccadic gain. I. The model and its biological substrate. *Biol. Cybern.* 75, 19–28. doi: 10.1007/BF00238736

Schweighofer, N., Doya, K., and Lay, F. (2001). Unsupervised learning of granule cell sparse codes enhances cerebellar adaptive control. *Neuroscience* 103, 35–50. doi: 10.1016/S0306-4522(00)00548-0

Schweighofer, N., Spoelstra, J., Arbib, M., and Kawato, M. (1998). Role of the cerebellum in reaching movements in humans. II. A neural model of the intermediate cerebellum. *Eur. J. Neurosci.* 10, 95–105. doi: 10.1046/j.1460-9568.1998.00007.x

Shadmehr, R., Smith, M. A., and Krakauer, J. W. (2010). Error correction, sensory prediction, and adaptation in motor control. *Annu. Rev. Neurosci.* 33, 89–108. doi: 10.1146/annurev-neuro-060909-153135

Silver, R. A., Cull-Candy, S. G., and Takahashi, T. (1996). {N}on-{NMDA} glutamate receptor occupancy and open probability at a rat cerebellar synapse with single and multiple release sites. *J. Physiol.* 494(Pt 1), 231–250. doi: 10.1113/jphysiol.1996.sp021487

Solinas, S., Nieus, T., and D'Angelo, E. (2010). A realistic large-scale model of the cerebellum granular layer predicts circuit spatio-temporal filtering properties. *Front. Cell. Neurosci.* 4:12. doi: 10.3389/fncel.2010.00012

Soman, K. P., Diwakar, S., and Ajay, V. (2005). *Insight Into Data Mining: Theory And Practice*. New Delhi (India): Prentice Hall India.

Standring, S. (2016). *Gray's Anatomy -The Anatomical Basis of Clinical Practice*. London: King's College London. doi: 10.1017/CBO9781107415324.004

Stock, A.-K., Gohil, K., Huster, R. J., and Beste, C. (2017). On the effects of multimodal information integration in multitasking. *Sci. Rep.* 7:4927. doi: 10.1038/s41598-017-04828-w

Tanaka, H., Ishikawa, T., Lee, J., and Kakei, S. (2020). The cerebro-cerebellum as a locus of forward model: A review. *Front. Syst. Neurosci.* 14:19. doi: 10.3389/fnsys.2020.00019

Tanaka, Y., Ohata, Y., Kawamoto, T., and Hirata, Y. (2010). Adaptive control of 2-wheeled balancing robot by cerebellar neuronal network model. *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society EMBC'* (Buenos Aires), 10, 1589–1592. doi: 10.1109/IEMBS.2010.5626673

Tartaglia, E. M., Brunel, N., and Mongillo, G. (2015). Modulation of network excitability by persistent activity: How working memory affects the response to incoming stimuli. *PLoS Comput. Biol.* 11, 1–27. doi: 10.1371/journal.pcbi.1004059

Thabtah, F. (2017). "Autism Spectrum Disorder Screening: Machine Learning Adaptation and DSM-5 Fulfillment," in *Proceedings of the 1st International Conference on Medical and Health Informatics*, (Taiwan: ACM), 1–6. doi: 10.1145/3107514.3107515

Thach, W. T. (1998). What is the role of the cerebellum in motor learning and cognition? *Trends Cogn Sci.* 2, 331–337. doi: 10.1016/S1364-6613(98)01223-6

Tseng, Y., Diedrichsen, J., Krakauer, J. W., Shadmehr, R., and Bastian, A. J. (2007). Sensory prediction errors drive cerebellum-dependent adaptation of reaching. *J. Neurophysiol.* 98, 54–62. doi: 10.1152/jn.00266.2007

Varshney, L. R., Sjöström, P. J., and Chklovskii, D. B. (2006). Optimal information storage in noisy synapses under resource constraints. *Neuron* 52, 409–423. doi: 10.1016/j.neuron.2006.10.017

Vijayan, A., Medini, C., Palolithazhe, A., Muralidharan, B., Nair, B., Diwakar, S., et al. (2015). "Modeling Pattern Abstraction in Cerebellum and Estimation of Optimal Storage Capacity," in *Proceedings of the Fourth International Conference on Advances in Computing, Communications and Informatics (ICACCI-2015)*, (Kochi: IEEE), 335–347. [pii] doi: 10.1109/ICACCI.2015.7275622

Vijayan, A., Nutakki, C., Kumar, D., Achuthan, K., Nair, B., and Diwakar, S. (2017). Enabling a freely accessible open source remotely controlled robotic articulator with a neuro-inspired control algorithm. *Int. J. Online Eng.* 13, 61–75. doi: 10.3991/ijoe.v13i01.6288

Vijayan, A., Singanamala, H., Nair, B., Medini, C., Nutakki, C., and Diwakar, S. (2013). "Classification of robotic arm movement using SVM and Naïve Bayes classifiers," in *Proceedings of the Innovative Computing Technology (INTECH), 2013 Third International Conference on* (IEEE), Piscataway. 263–268. doi: 10.1109/INTECH.2013.6653628

Wagner, M. J., Hyun Kim, T., Savall, J., Schnitzer, M. J., and Luo, L. (2017). Cerebellar granule cells encode the expectation of reward. *Nat. Lett.* 544, 96-100. doi: 10.1038/nature21726

Wilkins, A. (2017). Cerebellar dysfunction in multiple sclerosis. *Front. Neurol.* 8, 1–6. doi: 10.3389/fneur.2017.00312

Witten, I. H., Frank, E., and Hall, M. A. (2011). *Data Mining: Practical Machine Learning Tools and Techniques, Third Edition (The Morgan Kaufmann Series in Data Management Systems)*.Burlington Morgan Kaufmann.

Wolpert, D. M., and Kawato, M. (1998). Multiple paired forward and inverse models for motor control. *Neural Netw.* 11, 1317–1329. doi: 10.1016/S0893-6080(98)00066-5

Wolpert, D. M., Miall, R. C., and Kawato, M. (1998). Internal models in the cerebellum. *Trends Cogn. Sci.* 2, 338–347. doi: 10.1016/S1364-6613(98)01221-2

Yamaura, H., Igarashi, J., and Yamazaki, T. (2020). Simulation of a human-scale cerebellar network model on the K computer. *Front. Neuroinform.* 14:16. doi: 10.3389/fninf.2020.00016

Yamazaki, T., and Igarashi, J. (2013). Realtime cerebellum: A large-scale spiking network model of the cerebellum that runs in realtime using a graphics processing unit. *Neural Netw.* 47, 103–111. doi: 10.1016/j.neunet.2013.01.019

Yamazaki, T., and Lennon, W. (2019). Revisiting a theory of cerebellar cortex. *Neurosci. Res.* 148, 1–8. doi: 10.1016/j.neures.2019.03.001

Yamazaki, T., Igarashi, J., Makino, J., and Ebisuzaki, T. (2019). Real-time simulation of a cat-scale artificial cerebellum on PEZY-SC processors. *Int. J. High Perform. Comput. Appl.* 33, 155–168. doi: 10.1177/1094342017710705

Yamins, D. L. K., and DiCarlo, J. J. (2016). Using goal-driven deep learning models to understand sensory cortex. *Nat. Neurosci.* 19, 356–365. doi: 10.1038/nn.4244

Yang, Z., Zhong, S., Carass, A., Ying, S. H., and Prince, J. L. (2014). deep learning for cerebellar ataxia classification and functional score regression. *Mach. Learn. Med. Imaging* 8679, 68–76. doi: 10.1007/978-3-319-10581-9_9