Frequency-Dependent Synaptic Plasticity Model for Neurocomputing Applications

Abstract: In neuroscience, there is substantial evidence that suggests temporal filtering of stimulus by synaptic connections. In this paper, a novel frequency-dependent plasticity mechanism (FDSP) for neurocomputing applications is presented. It is proposed that synaptic junctions could be used to perform bandpass filtering on the input stimulus. The unique transfer function of a bandpass filter replaces the conventional weight value associated with synaptic connections. The proposed model has been simulated and rigorously tested with standard machine learning benchmarks such as XOR and multivariate IRIS dataset while utilising minimum resources. The proposed model offers a unique advantage and has the potential to overcome the burden of hidden layer neurons from the network. Exclusion of hidden layer from the network significantly reduces the size of the network and hence the computational effort required for classification tasks. The proposed FDSP mechanism allows for complete analogue system design with a frequency multiplexed communication scheme. The main goal of this study is to establish frequency-dependent plasticity as an alternative to existing time-domain based techniques. The proposed method has a number of applications in neurocomputing, low power IoT devices and compute-efficient Deep Convolutional Neural Networks (DCNNs).

Keywords: Machine learning; Neural networks; IoT devices, Neural engineering; Data classification; Synaptic plasticity.

1 Introduction

How does brain process information? No one can provide a complete and undisputed answer to this question as no one has actually created it. All that could be done is to observe how the brain processes information and provide different interpretations about it. The observations may be challenged. Despite the introduction of different brain signals recording techniques such as Functional Magnetic Resonance Imaging (fMRI), Electroencephalogram (EEG), and intracellular recording, the complete comprehension of the functioning of the brain at a macro level is still in early stages. Currently, there is a plethora of literature available that explains how actual biological systems function at the micro or cellular level (Dayan and Abbott, 2014). Hodgkin and Huxley (HH) were the first to propose an accurate neuron model in terms of the fourth-order dynamical system (Hodgkin and Huxley, 1952) referred to as HH model. There are a number of excitability and bursting behaviours that are attributed towards HH model. HH model is computationally inefficient as it represents a fourth-order dynamical system. A number of reductions for the original HH model have been proposed (FitzHugh, 1961; Nagumo, et al., 1962; Morris and Lecar, 1981; Izhikevich, 2003). These low dimensional models attempt to generate the similar neuronal dynamics that were previously achieved by complex HH model. The most influential model amongst all is the Morris-Lecar (ML) model (Morris and Lecar, 1981) which reduced the fourth-order HH model into a second-order system. In general, the neurons within a cortex can be classified into three classes on the basis of HH model which are Class I, II and III (Hodgkin, 1948). Most of the neurons in the cortex comprises of Class I and II neurons (Izhikevich, 2014). The characteristic curves for Class I type neuron is shown in Fig 1.

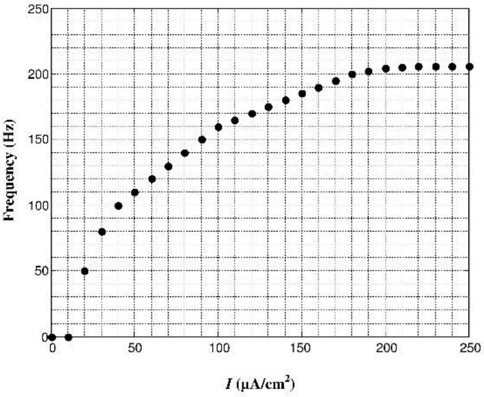

Fig. 1. The characteristic curve for Class I neuron model.

The input current is plotted on the x-axis and the output spiking frequency is plotted on the y-axis. As shown in Fig. 1, Class I neurons have a sub-linear curve with a threshold value that represents the encoding of an input stimulus into output spiking frequency. Whereas Class II neurons encode stimulus in more binary fashion by having no output for a lower range of input values and having near-constant output frequency for higher range of input. As evident from the characteristic curve of Class I neuron is the rate encoding feature of biological neurons. The stimulus applied to different classes of the neuron is encoded in terms of output spiking frequency. The frequency of output spikes may be interchanged into time lapse between two spikes. This time-lapse is generally referred to as Inter-Spike Interval (ISI). Related studies (Oswald, et al., 2007; Kepecs and Lisman, 2003) suggest that the ISI is considered as an important neuronal parameter for information encoding. The information regarding the stimulus is encoded into an output wave by varying the ISI parameter. The relationship between ISI (t_{ISI}) and output spiking frequency f is inversely proportional. The ISI for a burst of spikes can interchangeably be referred to as output spiking frequency of rate encoding neuron. This encoded frequency should be decoded at the receiving neuron to evaluate the actual stimulus applied at the presynaptic neuron.

As synaptic plasticity plays a key role in neuronal processing, in particular, from a neuro-computational perspective, an efficient plasticity mechanism must support the decoding of rate encoded information at the postsynaptic side. The Hebbian theory (Hebb, 1949) gained primary attention by the neurocomputational research community working in the field of Artificial Neural Networks (ANNs). This theory provides a mechanism for embedding plasticity in neural networks. The Hebbian theory states that the correlated spiking activity between the presynaptic and postsynaptic neuron strengthens the synaptic strength between the two neurons. This stimulus-driven change in synaptic efficacy is considered to be responsible for short-term memory (Freeman, 1995). Similarly, in neurocomputational applications, a weight value is normally used to represent the synaptic strength (Haykin, 1999). Almost all ANNs are based on weight values, considered as synaptic connections. Learning algorithms have been applied on ANNs to modify these weight values whereas the weight value represents a linear function between zero and a maximum value that is normally considered as one. In retrospect, zero represents a full disconnect between the presynaptic and postsynaptic neuron whereas one value represents a complete connection between the two neurons. Any intermediate value between zero and one represents the relative strength of the connection. If the weight value is incremented monotonically with time then the weight value can be represented as a linear function.

The paradigm of ANNs represents a numerical simulation model that runs on a sequential processor. The last two decades have witnessed a rapid growth in parallel implementation of ANNs utilising different platforms such as Clusters, GPUs and Multicore processors (Seiffert, 2004). The most predominant technique is to design a neural network on dedicated electronic hardware, termed as neuromorphic hardware (Mead, 1989).

Spike Time Dependent Plasticity (STDP) (Song, et al., 2000; Bi and Poo, 1998) has emerged as the most popular plasticity mechanism for neuromorphic hardware. There are a number of research groups currently working to embed plasticity in neuromorphic hardware through STDP (Jin et al., 2010; Schemmel et al., 2006, Indiveri, et al., 2006; Jo et al., 2010; Afifi, et al., 2009). STDP has its own advantages and disadvantages for hardware implementation. Disadvantages may include the inclusion of additional correlation detection circuitry as used in (Schemmel et al., 2006), implementation of weight value with 4-bit memory which provides a resolution of 16 possible weight values (Pfeil et al., 2012), and offline learning algorithms. This paper offers an alternative technique, named as Frequency-Dependent Synaptic Plasticity (FDSP) which is not entirely based on Hebbian theory. It attempts to contribute and suggest an out of the box solution for neurocomputational applications. Furthermore, it is explained in the following sections that a synaptic junction based on FDSP model has a non-linear transfer function which facilitates the classification of non-linear problems as compared to a network based on linear weight values.

The computational burden in the proposed work is on rather simple synaptic connections and dendrites. In the proposed architecture, the synapses perform bandpass filtering on inputs received from the axon of the presynaptic neuron. Whereas the dendrites of postsynaptic neurons perform integration on the synapse output and adder is employed where it is required to connect multiple inputs to a single neuron. In the proposed network, neurons are used to perform encoding. Therefore, the compute-intensive tasks such as filtering and integration are performed by the connections between the presynaptic and postsynaptic neuron. It is also in line with the biological evidence which states that the connection between the presynaptic and postsynaptic neuron performs major computation tasks. There are some interesting areas of the machine learning (Cui et al., 2019; Wang et al., 2019; Cai et al., 2019a; Cai et al., 2019b) where the proposed model may be employed.

2 FDSP Model

There is ample evidence that suggests that synaptic plasticity can be realised as temporal filtering by synapses. Markram states that the Hebbian theory, the most sought after theory by the neurocomputational research community, is an incomplete explanation of synaptic plasticity (Markram et al., 1998a). Markram further elaborates that during the formulation of famous Hebbian theory, Hebb overlooked some of the existing literature at that time which indicates the variation in synaptic response to different stimuli (Feng, 1941; Hutter, 1952; Liley and North, 1953; del Castillo and Katz, 1954; Liley, 1956). Despite the strong evidence put forward by Markram in favour of Frequency-Dependent Plasticity, there has not been a significant effort by the research community to adopt the same plasticity models for ANNs. It has been established (Markram et al., 1998b) that short-term synaptic plasticity performs temporal filtering on input feed to the synaptic junction. Synaptic junction under depression acts as a low-pass filter and attenuates high-frequency presynaptic firing whereas a synaptic junction under potentiation acts as a high-pass filter as it attenuates low-frequency presynaptic firing. As a result, a synapse that exhibits both short-term depression and potentiation acts as a bandpass filter.

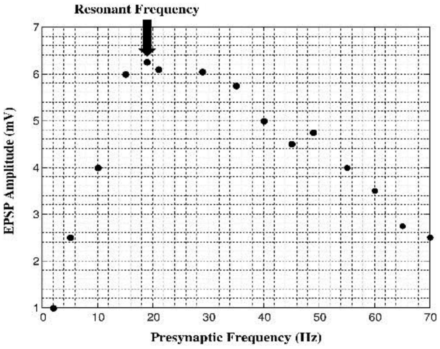

Fig. 2. Filtering operations at the synaptic level (Lisman, 1997). A synapse exhibiting both short term potentiation and depression acts as a bandpass filter.

In this context where synaptic junction functions as bandpass filters, a resonant frequency is associated with each synaptic junction. This resonant frequency may be different for different synaptic junctions associated with the same axon of a presynaptic neuron (Markram et al., 1998b; Lisman, 1997; Gupta, 2000). This selective filtering by the synaptic junctions facilitates selective triggering at the postsynaptic side on the basis of distinct frequency spikes generated by presynaptic neurons.

Apart from Markram, Izhikevich also provides strong evidence regarding selective triggering attained through Frequency-Dependent Plasticity (Izhikevich et al., 2003) in the cortex. It is argued that the synaptic junctions acting as bandpass filters facilitate selective communication between presynaptic and postsynaptic neurons which solely depends on the resonance feature of spikes generated by the presynaptic neuron (Fig. 2). The depiction for selective communication via resonance is shown in Fig 3.

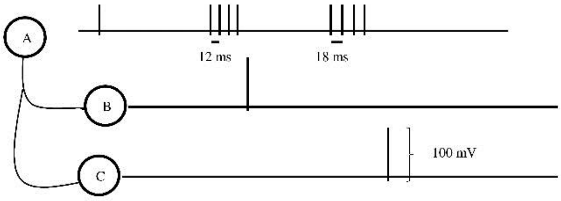

Fig. 3. A scenario to depict selective communication via resonance.
.

In Fig. 3, presynaptic neuron ‘A’ communicates selectively to postsynaptic neuron ‘B’ and ‘C’ via resonance. A *12 ms* resonance output is considered as resonant for postsynaptic neuron ‘B’ whereas *18 ms* resonance output is considered as resonant for postsynaptic neuron ‘C’. Alternatively, it can be stated that the presynaptic neuron ‘A’ generates spikes with a resonance of *12ms* which triggers only postsynaptic neuron ‘B’. Whereas when presynaptic neuron ‘A’ generates spikes with the resonance of *18ms*, it will only trigger postsynaptic neuron ‘C’. All these evidence suggest that synaptic junctions may act as bandpass filters facilitating selective communication between presynaptic and postsynaptic neurons.

The substantial evidence presented above from the neurophysiological studies suggest that plasticity solely depends on temporal filtering of spikes generated from the presynaptic neurons. Hence synaptic efficacy can be expressed in terms of frequency bandwidth and not necessarily with linearly scaled strength. It has been reported (Markram et al., 1998a) that the resonant frequency is unique for distinct synaptic junctions connected with the same axonal link. This unique resonant frequency is believed to play a key role in plasticity in biological neural networks. Frequency-dependent changes also play a key role in N-methyl D-aspartate receptor (NMDAR)-dependent synaptic plasticity, as suggested by (Kumar and Mehta, 2011), which is considered to play a key role in different learning processes within the cortex. From the above discussion, it is evident that the interpretation of synaptic plasticity is more biologically plausible in terms of frequency dependence as compared to the linear strength values represented as weight.

The depiction of a model that comprises of single axon connected to synaptic connection with distinct resonant frequency along with its biological counterpart is shown in Fig. 4 a and b.

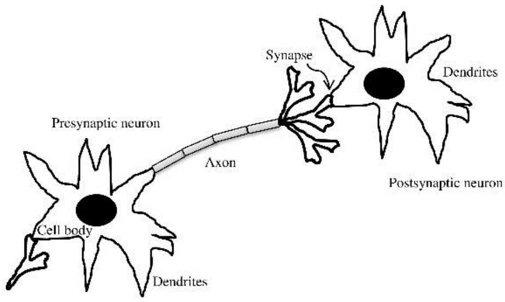

Fig. 4.a. A biological synaptic connection between pre and postsynaptic neuron

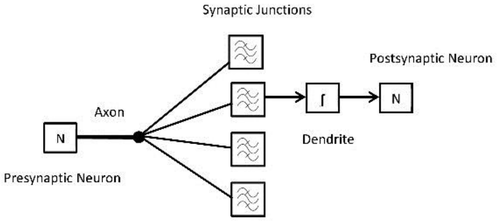

Fig. 4.b. Frequency-Dependent Synaptic Plasticity (FDSP) model

As shown in Fig. 4.b, synaptic junctions in the proposed model are realised with a bandpass filter, each having a distinct resonant frequency. The dendritic connection is considered to perform integration operation on the output from the synaptic connection. The pre and the postsynaptic neuron are considered to perform the same rate-encoding as observed by class I type of neurons. Note that the thick line in the model (depicted in Fig. 4.b) represents an axon link similar to the one shown in the biological model in Fig. 4.a. In the biological model, the single axon link is then distributed into axon tips. Each axon tip is connected to a distinct synaptic junction. These synaptic junctions are realised by a bandpass filter in the proposed FDSP model. The dendrite of a post-synaptic neuron is considered to perform integration operation on the output of synaptic junctions (Tran-Van-Minh et al., 2015). The postsynaptic neuron is considered as of the same type as the presynaptic neuron. However, the output neuron may also be considered as a typical threshold-based neuron. The adoption of an output neuron model depends on the application that is under consideration. A detailed mathematical explanation for synaptic processing through different aforementioned modules is covered in the following section.

As reported in (Wong-Riley, 1989) that dendrites consume 60 percent of the overall energy in the brain which indicates that the dendrites consume more power than a soma during an information processing task within the brain. In general, dendrite refers to the connection between the presynaptic and the postsynaptic neurons (Urbanska et al., 2008). This connection can be further divided into axon of the presynaptic neuron, synaptic cleft and dendrite of the postsynaptic neuron. In the proposed model, synaptic cleft is assigned the utmost responsibility of processing by selectively filtering out spikes to pass through the network while attenuating others.

3 Mathematical Model for Synaptic Processing

It is helpful to consider a complete synaptic connection along with presynaptic and postsynaptic neuron before considering mathematical formulation for synaptic processing. A synaptic connection that is utilised in this paper is shown in Fig. 5.

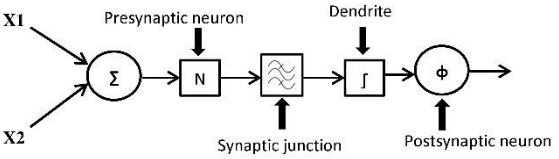

Fig. 5. A single synapse network

The single synaptic connection-based network model shown in Fig. 5 is a single connection representation of the network previously discussed in Fig. 4. It comprises of five modules, a summation unit, a rate encoding neuron labelled as ‘N’ which is based on the rate encoding feature of Class I neuron, bandpass filter to embed plasticity in the model which is realised as a synapse, an integrator unit to convert sinusoidal output from bandpass filter into a scalar value, realised as dendrite of the postsynaptic neuron and finally an output activation unit which is considered as the output or postsynaptic neuron in the proposed model. The adder units are included for accommodating multiple inputs to the same destination.

The adder unit is placed only when there are more than one inputs applied to the single input neuron. The operation at the accumulator unit can be expressed by a simple summation as shown in equation (1).

$$a = \sum_{i=1}^{n} x_i \qquad (1)$$

The output of the adder unit, ‘a’, is processed as an input to the rate encoding neuron that generates output sinusoidal wave with frequency *ω*. The main purpose of a rate encoding neuron is to convert stimulus value into output spiking rate. In order to test this hypothesis, a software model of Integrate-and-Fire neuron (Wagatsuma, 2017) is selected. The neuron threshold was set at ‘0.6’ which generates a sawtooth wave for stimulus above the threshold value. The expression for the neuron model is expressed by equation (2) and (3).

$$m(t) = \begin{cases} \frac{A}{2} - \frac{A}{\pi}\sum_{k=1}^{\infty}(-1)^k \frac{\sin(\omega k t)}{k} & \text{if a} > 0.6 \\ 0 & otherwise \end{cases} \qquad (2)$$

$$\omega = -3.6123 + (10.5824 \times a) \qquad (3)$$

Whereas A is the maximum amplitude of wave and k represents the threshold value. The values of constants shown in (3) are evaluated by using a linear regression technique. The neuron model is simulated in *MATLAB* environment. Input and output values are noted and constants are calculated. The output sinusoidal wave, *m(t)*, from the neuron, is applied to the bandpass filter. The output of the bandpass filter is expressed in equation (4).

$$g(t) = \begin{cases} m(t) & \text{if } \omega_l < \omega < \omega_h \\ \frac{m(t)}{\sqrt{2}M} & otherwise \end{cases} \qquad (4)$$

Here *g(t)* = *m(t)* if the frequency *ω* of *m(t)* falls in the passband of the bandpass filter where passband is spread from the lower cutoff frequency, ω_l, and the upper cutoff frequency, ω_h. Otherwise, the output of filter *g(t)* gets attenuated by a factor $m(t) = \sqrt{(2)}M$, where *M* is a scaling factor that depends on the width of the transition band of the bandpass filter. The output of the bandpass filter is integrated at the integrator unit according to equation (5).

$$\text{w} = \int g(t)dt \qquad (5)$$

The output from the integrator unit, *w*, is applied to the postsynaptic neuron. The postsynaptic neuron comprises a threshold unit which varies for different applications. The output of the hard-limiter threshold unit can be expressed by equation (6).

$$Y = \begin{cases} 1 & \text{if w} > \text{V}_T \\ 0 & otherwise \end{cases} \qquad (6)$$

Here V_T represents a threshold value. The above equations (2-6) mathematically express the processing by a single synaptic connection. A slight variation in this processing can be made as required by the application.

4 Learning

The synaptic junction may be expressed in the frequency domain as models perform bandpass filtering operation. The bandpass filter has a non-linear transfer function that can be expressed in the frequency domain as shown in Fig. 6.

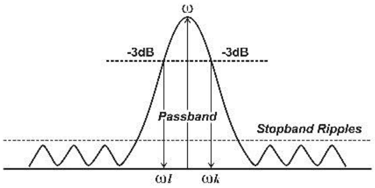

Fig. 6. The transfer function of a bandpass filter in the frequency domain.

The transfer curve shown for a bandpass filter has a center frequency, *ω*, two cutoff frequency points at -3db, lower cutoff frequency, ω_l and upper cutoff frequency point, ω_k. The bandwidth of a filter is defined between the lower and upper cutoff frequency points. Any frequency that falls within the bandwidth of bandpass filter is considered as passband frequency and appears on the fan-out of a bandpass filter without attenuation. All other frequencies are referred to as stopband frequencies and are significantly attenuated by the bandpass filter. A typical weight value which represents a singular scalar value may be adjusted to gain different efficacy for a synaptic connection. In the case of a tunable bandpass filter, the center frequency can be modified to update the efficacy of a synaptic connection in frequency domain considering a constant bandwidth of a filter. In case where variable bandwidth is desired, both values of lower cutoff frequency and upper cutoff frequency should be provided for defining synaptic efficacy.

In the proposed FDSP model, learning is currently implemented offline by evaluating the two cutoff frequencies that are lower cutoff frequency and the higher cutoff frequency. The evaluation of these two cutoff frequency points is relatively easy to interpret. Consider a vector *X* which represents input feature vector associated with a class C_i. It is considered here that the synaptic connection should allow all feature values associated with a particular class C_i while blocking all other values. The relationship between the feature vector and the associated class can be expressed by equation (7).

$$f[X] = [\, x_1, x_2, x_3, x_4, \ldots\ldots x_i \ldots\ldots\ x_n] \ \epsilon\ C_i \qquad (7)$$

The minimum and maximum values from the feature set associated with a particular class can be evaluated by using dedicated functions such as shown in equation (8) and (9)

$$b_1 = \min(f[X]) \qquad (8)$$

$$b_2 = \max(f[X]) \qquad (9)$$

The values of b_1 and b_2 can be used in equation (3) in place of *a* to evaluate the frequency value for lower and upper cutoff frequency points such as shown in equation (10) and (11).

$$\omega_l = -3.6123 + (10.5824 \times b_1) \qquad (10)$$

And;

$$\omega_h = -3.6123 + (10.5824 \times b_2) \qquad (11)$$

These two cutoff frequency points will allow stimulus values associated with a particular class to pass through them while blocking others. The feature values are initially applied to a rate encoding neuron and encoded into frequencies in accordance with equation (2), the output wave from the rate neuron is a wave representing the original stimulus value applied to the neuron in terms of frequency.

There is one important assumption to the feature values expressed in equation (7) that they all belong to a single class C_i. It is now considered that the feature vector has intermixed values from two different classes, C_i and C_j as shown in equation (12) and (13).

$$f[X] = \left[\, x_1, x_2, x_3, \ldots x_i, x_j, x_{j+1}, x_{j+2}, x_k, x_{k+1}\ \ldots\ x_n \right] \qquad (12)$$

Where;

$$\begin{gathered} [\, x_1, x_2, x_3, x_4, \ldots\ldots x_i \,] \ \epsilon\ C_i \\ \left[\ x_j, x_{j+1}, x_{j+2} \,\right] \epsilon\ C_j \\ [\, x_k, x_{k+1}\ \ldots\ldots\ x_n] \ \epsilon\ C_i \end{gathered} \qquad (13)$$

The feature vector now has spatial discontinuity for class C_i as it now contains some intermediate values from class C_j. A single synapse cannot be employed for such distributed feature set containing values from different classes because the minimum and maximum functions cannot be applied to the feature vector as it contains values from other class as well. To resolve this problem, the feature vector must be divided into two subclasses C_i such as C_{i1} and C_{i2}. The updated feature vector will be represented by equation (14).

$$\begin{gathered} [\, x_1, x_2, x_3, x_4, \ldots\ldots x_i \,] \ \epsilon\ C_{i1} \\ \left[\ x_j, x_{j+1}, x_{j+2} \,\right] \epsilon\ C_j \\ [\, x_k, x_{k+1}\ \ldots\ldots\ x_n] \ \epsilon\ C_{i2} \end{gathered} \qquad (14)$$

These three feature vector requires three synaptic connections and learning can now be performed according to equations (8)-(11).

5 Benchmarking

i. XOR problem

In FDSP model, the XOR gate operation is performed by the network shown in Fig. 5.

The adder unit shown in Fig. 5 combines the two inputs and provides accumulated stimulus values to the rate neuron. The accumulated values for different input patterns are listed in Table I.

Here the binary value ‘0’ is encoded as ‘0.5’. Form Table I, it is evident that the adder unit provides three possible values of ‘1’, ‘1.5’ and ‘2’ for any combination of inputs. The accumulated stimuli value is then passed to the rate neuron which further encodes the applied value into output spiking frequency in accordance with (2).

TABLE I
STIMULUS ACCUMULATION AT ADDER UNIT

X1	X2	Accumulated Output
0.5	0.5	1
0.5	1.0	1.5
1.0	0.5	1.5
1.0	1.0	2

The encoding frequencies for the accumulated input are mentioned in Table II.

TABLE II
RATE ENCODING BY INPUT NEURON

Input to neuron	Output Frequency
1.0	ω_1
1.5	ω_2
2.0	ω_3

As stated in eq (3), ω_1 is equal to 6.9701 rad/s, ω_2 is equal to 17.5525 rad/s and ω_3 is equal to 28.1349 rad/s. These three frequencies distinctly identify the applied stimulus to the input neuron in terms of output spiking frequency.

The relationship between XOR output and the frequencies generated by encoding neuron can be summarised in Table III.

TABLE III
FREQUENCY-BASED TRUTH TABLE FOR XOR GATE

Output Frequency	XOR
ω_1	False
ω_2	True
ω_3	False

It is evident from Table III that the true condition for the XOR gate is signalled when frequency ω_2 is generated by the encoding input neuron. The frequency ω_2 corresponds to accumulated input value '1.5' applied to the rate encoding neuron. The accumulated value of '1.5' is obtained by adding two possible complimentary values of binary '0' and '1' applied at either of the input connections *X1* and *X2*.

The learning for XOR gate is followed in accordance with the mechanism explained in section 4. Since there is only one feature value for the true class for XOR gate therefore *min(x) = max(x) = 1.5* which corresponds to frequency ω_2. The bandpass filter is tuned for ω_2 as center passband frequency. By this synaptic modification, the network shown in Fig. 5 will behave as an XOR gate.

ii. IRIS plant classification

Iris plant dataset represents a classical problem of multivariate data classification. It comprises of feature vectors involving petal length, petal width, sepal length and sepal width. These four feature sets are used for classifying three different types of Iris plants namely setosa, virginica and versicolor. The dataset for this problem is obtained from UCl machine learning repository (Fisher, 1936). A network is considered to be capable of classifying multivariate data if it properly classifies the aforementioned three plants. The generic FDSP model for *N* input and *N* output multivariate data classification problem is shown in Fig. 7.

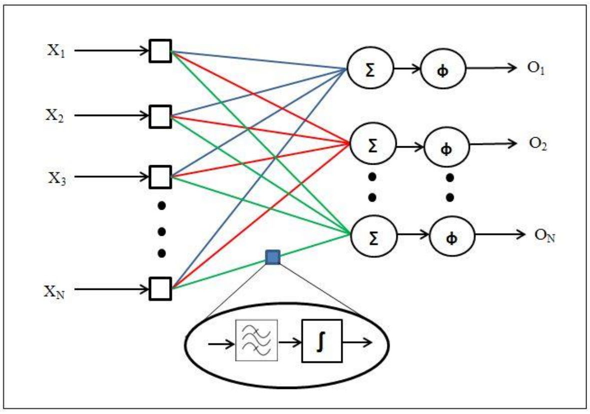

Fig. 7. Generic FDSP Model for multivariate data classification

The Iris plant classifier is a two-layer network comprising of input and output layer neurons. The input layer neuron is a rate encoding neuron explained in section III where the number of input layer neurons is equal to the number of variables in feature space. In IRIS classification, there are four feature vectors namely petal length, petal width, sepal length and sepal width assigned to input nodes *X1*, *X2*, *X3* and *X4*, respectively. The number of output neurons is proportional to the number of classes. There are three classes, representing three different flowers setosa, versicolor and virginica assigned to output nodes *O*1, *O*2 and *O*3 respectively. Each synaptic connection is realised with a bandpass filter and an integrator, shown with a small oval in Fig. 7.

The Iris plant dataset contains a total 150 samples which are equally divided into three classes, 50 samples for each class. There are four feature values for each sample. In order to implement learning according to FDSP mechanism, *min(x)* and *max(x)* is calculated over the set of 40 out of 50 values designated to each class. Ten values are kept for functional testing purpose. Table IV entails information regarding *max* and *min* feature values associated with each class.

TABLE IV
IRIS PLANT CLASS BOUNDARY VALUES

Class	Value	Feature 1	Feature 2	Feature 3	Feature 4
Setosa	Min.	4.3	2.3	1.1	0.1
	Max.	5.8	4.4	1.9	0.6
Versicolor	Min.	4.9	2.2	3.0	1.0
	Max.	7.0	3.4	5.1	1.8
Virginica	Min.	4.9	2.2	4.5	1.4
	Max.	7.7	3.8	6.9	2.5

The minimum and maximum values help to identify the cutoff frequency values for each synaptic connection according to the FDSP learning mechanism stated in section 4. The values of feature 4 are close to the threshold of the encoding neuron which is '0.6' therefore a bias value of '1' is added to the encoding neuron for feature 4. The feature *min* and *max* values along with their corresponding frequency value in radian per second is listed in Table V. The frequency values are evaluated using equation (3). The conversion of feature set values into rate-neuron encoding frequency values as shown in Table V help to identify the target frequencies for individual synaptic connections.

TABLE V
STIMULUS VALUE V/S FREQUENCY VALUE (IN RAD/S)

Feature 1	Freq.	Feature 2	Freq.	Feature 3	Freq.	Feature 4	Freq.
4.3	41.8	2.3	20.7	1.1	8.0	1.1	8.0
5.8	57.7	4.4	42.9	1.9	16.4	1.6	13.3
4.9	48.2	2.2	19.6	3.0	28.1	2.0	17.5
7.0	70.4	3.4	32.3	5.1	50.3	2.8	26.0
4.9	48.2	2.2	19.6	4.5	44.0	2.4	21.7
7.7	77.8	3.8	36.6	6.9	69.4	3.5	33.4

These frequency values provide lower and upper cutoff frequencies for synaptic connections in the network model as shown in Fig. 7. The bandwidth of each synaptic filter depends on the distance between the two cutoff frequency points. From Table V, the filter associated with a very first synaptic connection that is between the first feature *X1* and the first output neuron *O*1 has the lower cutoff frequency point set at 41.8 rad/s and the upper cutoff frequency point set at 57.7 rad/s. This will provide a total bandwidth of 15.9 rad/s for the said filter. Similarly, the bandwidth of each synaptic connection for the network shown in Fig. 7 can be summarized in Table VI.

TABLE VI
INDIVIDUAL SYNAPTIC CONNECTION BANDWIDTH

Input node	Output node	Lower cutoff Frequency	Upper cutoff Frequency	Bandwidth
x1	O1	41.8	57.7	15.9
x2	O1	20.7	42.9	22.2
x3	O1	8.0	16.4	8.4
x4	O1	8.0	13.3	5.3
x1	O2	48.2	70.4	22.2
x2	O2	19.6	32.3	12.7
x3	O2	28.1	50.3	22.2
x4	O2	17.5	26.0	8.5
x1	O3	48.2	77.8	29.6
x2	O3	19.6	36.6	17
x3	O3	44.0	69.4	25.4
x4	O3	21.7	33.4	11.7

Table VI provides information regarding variable bandwidth requirement for individual synaptic connection where the frequency value is in radian per second. Another important aspect is the overlapping between the passband frequencies associated with a single feature value and different output neurons. This overlapping will hinder the proper identification of a particular class at the output side. However, for setosa class, there are at least two feature values (3rd and 4th) which do not overlap with the same feature values of other classes. The main problem lies with the Frequency-based of versicolor and virginica as they have overlapping feature values in all four feature sets. This overlapping is responsible for most of the misclassification between the two classes. The misclassification has been briefly discussed in next section.

The output neuron model used for classification of Iris plant contains two units, adder and activation. The adder unit accumulates the values from all synaptic connections. It should be noted that the output from the integrator unit is labelled as 'w' in (5). The accumulation at the adder unit placed before the output neuron can be expressed by equation (15)

$$u_i = \sum_{i=1}^{n} w_i \qquad (15)$$

The output of the adder unit is fed to the activation unit of the postsynaptic neuron. The activation unit has a different transfer function as compared to the one described in equation (6). Here rather than specifying a constant threshold value a more dynamic function is used to determine output. The activation function used for this model may be expressed by equation (16).

$$Y_i = \begin{cases} 1 & \text{if } \max(u_1, u_2, u_3) == u_i \\ 0 & otherwise \end{cases} \qquad i = 1,2,3 \qquad (16)$$

It can be said that *Y=1* is generated for a node that provides maximum accumulated output from the adder unit. This output from the adder unit is only possible when all synaptic connections associated with a particular output node provide maximum integrated output from the integrator units. The integrator unit is associated with each synaptic link.

6 Simulations

Simulation for the FDSP models presented in section 5, have been carried out in *MATLAB* environment. In all *MATLAB* simulations, a modulus operator is inserted after each integrator block. The basic requirement for including the modulus operator is the phase shift property of a bandpass filter. Due to the phase shift, the integrator sometimes produces negative values in the *MATLAB* environment. In order to avoid negative values from inclusion into the overall result of the network, a modulus operator is placed in every synaptic connection after the integrator block. A model to implement XOR gate operations was designed similar to the one shown in Fig. 5. Neuron model was designed separately for Integrate-and-Fire neuron as mentioned in section 3. For analogue bandpass filter, Butterworth second-order bandpass filter (Butterworth, 1930) was selected. It was calculated that if the frequency by the encoding neuron falls in the passband of the bandpass filter then the output from the integrator unit is greater than scalar value '0.005'. Therefore, the threshold value for the comparator block is kept at '0.005'. The bandpass filter is adjusted in accordance with the values listed in Table III. For the implementation of XOR gate, ω_2 is selected as the center frequency. The test accuracy of XOR is 100 percent. An XOR gate implementation in the proposed model has such high accuracy due to the appropriate selection of frequency for the single synaptic connection, which is ω_2.

The Iris plant classifier implementation using the FDSP model is shown in Fig. 7. The same rate encoding neuron as mentioned in section 3 is utilised for encoding feature values. The bandpass filter adopted is a second-order Butterworth bandpass filter. The two cutoff frequency points are selected in accordance with values listed in Table V. The performance of the classifier can be evaluated with the help of a confusion matrix. The confusion matrix for the Iris plant classifier is shown as Table VII.

TABLE VII
CONFUSION MATRIX FOR IRIS PLANT

		Observed			
	n=150	Setosa	Versicolor	Virginica	Correct output
Correct	Setosa	50	0	0	50
	Versicolor	0	37	13	37
	Virginica	1	7	42	42
	Total correct output out of 150 samples				129

The above confusion matrix represents that classifier performed with 86% accuracy in Iris plant classification task. Note that the Setosa plant has the best classification accuracy of 100%. The main problem lies with the classification between versicolor and virginica plants. The input features for these two plants overlap for all the four feature sets. On the other hand, setosa has at least two feature values (3rd and 4th) which do not overlap with feature values of other classes. The accuracy of the network is increased by minimising the overlap between feature values of versicolor and virginica. The 4th feature vector of both these plants has been selected. Considering the 4th feature value, the overlap for both classes' ranges from '1.4' to '1.8'. The upper cutoff point evaluated for versicolour is evaluated at 1.8 whereas the lower cutoff frequency point evaluated for virginica is at '1.4'. If '1.6' evaluated as the center point between '1.4' and '1.8' then the upper cutoff frequency for versicolor can be evaluated at '1.6' and the lower cutoff frequency for virginica may also be evaluated at '1.6'. This will reduce the overlap between the two classes for the 4th feature value. Similarly, the 3rd feature vector has a range of overlapping values from '4.5' to '5.1' between classes versicolor and virginica. If this overlapping range is assigned to virginica then the upper cutoff frequency point for versicolor is evaluated at '4.5' and the lower cutoff frequency point for virginica is also evaluated at '4.5'. The confusion matrix shown as Table VIII after simulating updated network configuration is as under:

TABLE VIII
REVISED CONFUSION MATRIX FOR IRIS PLANT

		Observed			
	n=150	Setosa	Versicolor	Virginica	Correct output
Correct	Setosa	50	0	0	50
	Versicolor	2	40	8	40
	Virginica	2	4	44	44
	Total correct output out of 150 samples				134

The accuracy of the IRIS classifier is increased to '89.33' percent. Further modifications such as increasing the order of the bandpass filter will also facilitate in increasing the accuracy of the network. Also a bias value of '1' is added to the 4th feature vector values. The said values are below the threshold of the neuron which is '0.6'. Adding bias to the feature values will provide the necessary offset to push all the feature values above the threshold point of encoding neuron.

7 Discussion

The FDSP model for Iris plant classification problem is compared with the existing neuromorphic and neural network-based models. The comparison is shown in Table IX.

TABLE IX
COMPARISON TABLE FOR IRIS PLANT (VALUES ADOPTED FROM (GHANI ET AL., 2012))

Algorithm	Encoding neurons	Inputs	Hidden	Output	Total neurons	Accuracy (%)
Spike Prop (Bohte, et al., 2002)	50		10	03	63	96.1
Wu et al. (Wu et al., 2006)	09		06	01	16	96.6
SRM based SNN (Belatreche, et al., 2006)	16		10	01	27	97.3
Dynamic synapse based SNN Belatreche et al. (Belatreche, et al., 2006)		04	10	01	15	96.0
Matlab BP		04	10	03	17	94.8
Matlab LM		04	10	03	17	94.7
Matlab RP		04	10	03	17	94.73
Spiking Synapse Hardware Weight Adaption (Ghani et al., 2012)		04	06	01	11	95.0
FDSP Model	**04**		**0**	**03**	**07**	**89.33**

As shown in Table IX, there exists a significant advantage of using the FDSP model. The main advantage is the exclusion of the hidden layer from the network. This exclusion of the hidden layer reduces the number of synaptic connections required for the same task. This will, in turn, reduce the computational burden at the system level.

To the best of author's knowledge, the presented model supersedes the existing neural network models in terms of computational efficiency by completely avoiding hidden layer for classification of non-linearly separable tasks. This omission of the hidden layer from the network, in turn, reduces the number of interconnection required for the network. In retrospect, it directly impacts the number of weight values required by the network to train. In the proposed model each input node is connected to every output node by a synaptic connection which provides non-linear response desirable for neural computation. In the authors' most recent work (Khan, et al., 2017) an area-efficient hardware implementation of population coding using the synaptic model has been demonstrated. This population coding model exhibits the noise-tolerant and efficient implementation of the neural network for decoding sensory information through neuromorphic hardware.

8 Conclusion

In this paper, a novel Frequency-Dependent Synaptic Plasticity (FDSP) mechanism is presented as an alternative approach to embed plasticity in neural networks. The synaptic plasticity is realised by placing a bandpass filter as a synaptic connection. The bandpass filter replaces the weight value that is associated with previously mentioned neural network models to represent synaptic strength. Two standard benchmark problems such as an XOR gate and Iris plant have been simulated and tested for the presented FDSP model. It has been demonstrated that the models which may only be implemented through multilayer perceptron could be implemented by a single synaptic connection through proposed FDSP model without employing complex learning algorithms. The learning algorithm is currently performed offline with the simplest function of calculating the minimum and maximum from the feature vector associated with a particular class.

In multivariate data classification, it has been shown that the classification is possible with only input and output layers without incorporating any hidden layer neurons. This is the most compressed implementation that involves the least number of neurons that can be implemented in any neural network for the classification task. This is primarily due to the inclusion of synaptic function that is nonlinear in nature. As compute time and power consumption are one of the key factors in low power IoT devices, front-end signal conditioning and DCNNs, authors envisage a number of applications in applied machine learning and low power IoT devices that could potentially benefit from the proposed method.

References

Afifi, A., Ayatollahi, A. and Raissi, F. (2009). STDP implementation using memristive nanodevice in CMOS-Nano neuromorphic networks. *IEICE Electronics Express*, 6(3), pp.148-153.

Belatreche, A., Maguire, L. and McGinnity, M. (2006). Advances in Design and Application of Spiking Neural Networks. *Soft Computing*, 11(3), pp.239-248.

Bi, G. and Poo, M. (1998). Synaptic Modifications in Cultured Hippocampal Neurons: Dependence on Spike Timing, Synaptic Strength, and Postsynaptic Cell Type. *The Journal of Neuroscience*, 18(24), pp.10464-10472.

Bohte, S., Kok, J. and La Poutré, H. (2002). Error-backpropagation in temporally encoded networks of spiking neurons. *Neurocomputing*, 48(1-4), pp.17-37.

Butterworth, S. (1930) 'On the theory of filter amplifiers', *Wireless Engineer*, *7*(6), pp.536-541.

Cai, X., Niu, Y., Geng, S., Zhang, J., Cui, Z., Li, J. and Chen, J. (2019a). An under-sampled software defect prediction method based on hybrid multi-objective cuckoo search. *Concurrency and Computation: Practice and Experience*.

Cai, X., Geng, S., Wu, D., Wang, L. and Wu, Q. (2019b). A unified heuristic bat algorithm to optimize the LEACH protocol. Concurrency and Computation: Practice and Experience.

Cui, Z., Du, L., Wang, P., Cai, X. and Zhang, W. (2019). Malicious code detection based on CNNs and multi-objective algorithm. *Journal of Parallel and Distributed Computing*, 129, pp.50-58.

Dayan, P. and Abbott, L. (2014). *Theoretical Neuroscience*. Cambridge: MIT Press.

del Castillo, J. and Katz, B. (1954). Statistical factors involved in neuromuscular facilitation and depression. *The Journal of Physiology*, 124(3), pp.574-585.

Feng, T.P., (1941) 'Studies on the neuromuscular junction. XXVI. The changes of the end-plate potential during and after prolonged stimulation', *Chinese Journal of Physiology*, *16*, pp.341-372.

Fisher, R. (1936). The Use of Multiple Measurements in Taxonomic Problems. *Annals of Eugenics*, 7(2), pp.179-188.

FitzHugh, R. (1961). Impulses and Physiological States in Theoretical Models of Nerve Membrane. *Biophysical Journal*, 1(6), pp.445-466.

Freeman, W. (1995). The Hebbian paradigm reintegrated: Local reverberations as internal representations. *Behavioral and Brain Sciences*, 18(4), pp.631-631.

Ghani, A., McDaid, L., Belatreche, A., Hall, S., Huang, S., Marsland, J., Dowrick, T. and Smith, A. (2012). Evaluating the generalisation capability of a CMOS based synapse. *Neurocomputing*, 83, pp.188-197.

Gupta, A. (2000). Organizing Principles for a Diversity of GABAergic Interneurons and Synapses in the Neocortex. *Science*, 287(5451), pp.273-278.

Hodgkin, A. (1948). The local electric changes associated with repetitive action in a non-medullated axon. *The Journal of Physiology*, 107(2), pp.165-181.

Hebb, D. (1949) *The organization of behavior*, 1st ed., Wiley, New York.

Hodgkin, A. and Huxley, A. (1952). A quantitative description of membrane current and its application to conduction and excitation in nerve. *The Journal of Physiology*, 117(4), pp.500-544.

Haykin, S. (1999). *Neural networks*. Upper Saddle River, N.J.: Prentice Hall.

Hutter, O. (1952). Post-tetanic restoration of neuromuscular transmission blocked by d-tubocurarine. *The Journal of Physiology*, 118(2), pp.216-227.

Izhikevich, E. (2003). Simple model of spiking neurons. *IEEE Transactions on Neural Networks*, 14(6), pp.1569-1572.

Indiveri, G., Chicca, E. and Douglas, R. (2006). A VLSI Array of Low-Power Spiking Neurons and Bistable Synapses With Spike-Timing Dependent Plasticity. *IEEE Transactions on Neural Networks*, 17(1), pp.211-221.

Izhikevich, E., Desai, N., Walcott, E. and Hoppensteadt, F. (2003). Bursts as a unit of neural information: selective communication via resonance. *Trends in Neurosciences*, 26(3), pp.161-167.

Izhikevich, E. (2014). *Dynamical Systems in Neuroscience*. Cambridge: MIT Press.

Jin, X., Rast, A., Galluppi, F., Davies, S. and Furber, S., (2010), 'Implementing spike-timing-dependent plasticity on SpiNNaker neuromorphic hardware' In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1-8, IEEE.

Jo, S., Chang, T., Ebong, I., Bhadviya, B., Mazumder, P. and Lu, W. (2010). Nanoscale Memristor Device as Synapse in Neuromorphic Systems. *Nano Letters*, 10(4), pp.1297-1301.

Kepecs, A. and Lisman, J. (2003). Information encoding and computation with spikes and bursts. *Network: Computation in Neural Systems*, 14(1), pp.103-118.

Khan, S., Ghani, A. and Khurram, M. (2017). Population coding for neuromorphic hardware. *Neurocomputing*, 239, pp.153-164.

Kumar, A. and Mehta, M. (2011). Frequency-Dependent Changes in NMDAR-Dependent Synaptic Plasticity. *Frontiers in Computational Neuroscience*, 5.

Liley, A. and North, K. (1953). An Electrical Investigation of Effects Of Repetitive Stimulation on Mammalian Neuromuscular Junction. *Journal of Neurophysiology*, 16(5), pp.509-527.

Liley, A. (1956). The quantal components of the mammalian end-plate potential. *The Journal of Physiology*, 133(3), pp.571-587.

Lisman, J. (1997). Bursts as a unit of neural information: making unreliable synapses reliable. *Trends in Neurosciences*, 20(1), pp.38-43.

Markram, H., Gupta, A., Uziel, A., Wang, Y. and Tsodyks, M. (1998a). Information Processing with Frequency-Dependent Synaptic Connections. *Neurobiology of Learning and Memory*, 70(1-2), pp.101-112.

Markram, H., Wang, Y. and Tsodyks, M., (1998b), 'Differential signaling via the same axon of neocortical pyramidal neurons' In *Proceedings of the National Academy of Sciences*, *95*(9), pp.5323-5328.

Morris, C. and Lecar, H. (1981). Voltage oscillations in the barnacle giant muscle fiber. *Biophysical Journal*, 35(1), pp.193-213.

Mead, C. (1989). *Analogue VLSI and neural systems*, 1st ed. Reading, Mass.: Addison-Wesley.

Nagumo, J., Arimoto, S. and Yoshizawa, S. (1962). An Active Pulse Transmission Line Simulating Nerve Axon. *Proceedings of the IRE*, 50(10), pp.2061-2070.

Oswald, A., Doiron, B. and Maler, L. (2007). Interval Coding. I. Burst Interspike Intervals as Indicators of Stimulus Intensity. *Journal of Neurophysiology*, 97(4), pp.2731-2743.

Pfeil, T., Potjans, T., Schrader, S., Potjans, W., Schemmel, J., Diesmann, M. and Meier, K. (2012). Is a 4-Bit Synaptic Weight Resolution Enough? – Constraints on Enabling Spike-Timing Dependent Plasticity in Neuromorphic Hardware. *Frontiers in Neuroscience*, 6.

Schemmel, J., Grubl, A., Meier, K. and Mueller, E., (2006), 'Implementing synaptic plasticity in a VLSI spiking neural network model' In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pp. 1-6, IEEE.

Seiffert, U. (2004). Artificial neural networks on massively parallel computer hardware. *Neurocomputing*, 57, pp.135-150.

Song, S., Miller, K. and Abbott, L. (2000). Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience*, 3(9), pp.919-926.

Tran-Van-Minh, A., Cazé, R., Abrahamsson, T., Cathala, L., Gutkin, B. and DiGregorio, D. (2015). Contribution of sublinear and supralinear dendritic integration to neuronal computations. *Frontiers in Cellular Neuroscience*, 9.

Urbanska, M., Blazejczyk, M. and Jaworski, J. (2008) 'Molecular basis of dendritic arborization', *Acta neurobiologiae experimentalis*, *68*(2), p.264.

Wagatsuma, H., Simulink Design - Leaky Integrate-and-Fire Neuron Model, Dynamic Brain. [online] https://dynamicbrain.neuroinf.jp / modules / xoonips / detail.php ? item_id = 6498 (Accessed 20 September 2017).

Wang, P., Huang, J., Cui, Z., Xie, L. and Chen, J. (2019). A Gaussian error correction multi-objective positioning model with NSGA-II. *Concurrency and Computation: Practice and Experience*.

Wong-Riley, M. (1989). Cytochrome oxidase: an endogenous metabolic marker for neuronal activity. *Trends in Neurosciences*, 12(3), pp.94-101.

Wu, Q., McGinnity, T., Maguire, L., Glackin, B. and Belatreche, A. (2006). Learning under weight constraints in networks of temporal encoding spiking neurons. *Neurocomputing*, 69(16-18), pp.1912-1922.