BIOLEAF: A BIO-PLAUSIBLE LEARNING FRAMEWORK FOR TRAINING SPIKING NEURAL NETWORKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Our brain consists of biological neurons encoding information through accurate spike timing, yet both the architecture and learning rules of our brain remain largely unknown. Comparing to the recent development of backpropagation-based (BP-based) methods that are able to train spiking neural networks (SNNs) with high accuracy, biologically plausible methods are still in their infancy. In this work, we wish to answer the question of whether it is possible to attain comparable accuracy of SNNs trained by BP-based rules with bio-plausible mechanisms. We propose a new bio-plausible learning framework, consisting of two components: a new architecture, and its supporting learning rules. With two types of cells and four types of synaptic connections, the proposed local microcircuit architecture can compute and propagate error signals through local feedback connections and support training of multi-layers SNNs with a globally defined spiking error function. Under our microcircuit architecture, we employ the Spike-Timing-Dependent-Plasticity (STDP) rule operating in local compartments to update synaptic weights and achieve supervised learning in a biologically plausible manner. Finally, We interpret the proposed framework from an optimization point of view and show the equivalence between it and the BP-based rules under a special circumstance. Our experiments show that the proposed framework demonstrates learning accuracy comparable to BP-based rules and may provide new insights on how learning is orchestrated in biological systems.

1 INTRODUCTION

Thanks to greater computing power, deep learning has gained remarkable achievements in recent years (Hinton et al., 2006; Bengio & LeCun, 2007; Schmidhuber, 2015; Goodfellow et al., 2016). However, learning by backpropagation (BP) (Rumelhart et al., 1986) is still the most popular method, which is generally believed impossible to be implemented in our brains (Illing et al., 2019). As compared to deep neural networks (DNNs), our brain, the only known true intelligence system, is more energy efficient (Von Neumann, 2012), robust (Denève et al., 2017; Qiao et al., 2019), and capable of achieving life-long learning (Parisi et al., 2019), online learning (Lobo et al., 2020), logic reasoning (Monti & Osherson, 2012), and has many other advantages (Raichle et al., 2001). The development of artificial intelligence (AI) may benefit from investing in how our brain works.

Our brain is a complex system consisting of neurons that communicate with each other through spikes. Therefore, people tried to use simplified spiking neurons to form a network that mimic the function of our brain. Such spiking neural networks (SNNs) can naturally exploit spatio-temporal data with each neuron's internal temporal dynamics (Yang et al., 2021), and save orders of magnitude of less energy when running on neuromorphic hardwares (Davies et al., 2018; Kim et al., 2020b; Davies et al., 2021). However, the training of SNNs is difficult.

Recent developments of the direct training methods of SNNs mainly diverge into two streams: BP-based rules and bio-plausible rules (Hao et al., 2020).

BP-based learning rules include: the activation-based surrogate gradient methods (Zenke & Ganguli, 2018; Shrestha & Orchard, 2018; Wu et al., 2018), the timing-based methods (Zhang & Li, 2020), the combination of both (Kim et al., 2020a), and the recently proposed neighborhood aggregation method - NA (Yang et al., 2021). These BP-based methods gained great performance improvement

and helped SNNs to be implemented on real-world problems, yet their biological plausibility remains unresolved: the co-existence of both forward and backward signals requires a neuron to fire two sets of uncorrelated signals from the same neuron body, which is not bio-plausible.

While the other branch, the bio-plausible learning rules, represented by the STDP (Taylor et al., 1973; Levy & Steward, 1983) and the Widrow-Hoff (WH) (Widrow & Hoff, 1960) rules, adjusts parameters using local plasticity only.

The STDP learning rule is built upon the Hebbian learning rule, which can be informally described as: "Cells that fire together, wire together" (Hebb, 1949). Following this rule, STDP adjusts synaptic weights by evaluating the timing correlation: If a presynaptic neuron fires a few milliseconds before a postsynaptic neuron, meaning this presynaptic spike contributes to the firing of the postsynaptic neuron, their connection is strengthened (causal), or called long-term potentiation. Whereas the opposite temporal order results in long-term depression (acausal). Although STDP demonstrates its potential usefulness in both supervised and unsupervised manners, it is unlikely that STDP works alone: Strengthened connection makes the firing activity of a pair of neurons more synchronized, and vice versa. Due to the existence of the positive feedback loop, one needs to introduce additional tricks to stabilize the learning process - such as winner-takes-all (WTA) (Nessler et al., 2013; Diehl & Cook, 2015; Kheradpisheh et al., 2018; Saunders et al., 2018), weights normalization (Ferré et al., 2018), weights clamping (Diehl & Cook, 2015; Kheradpisheh et al., 2018; Lee et al., 2018; Saunders et al., 2018), layer-by-layer training (Kheradpisheh et al., 2018), and others (Panda & Roy, 2016).

In comparison, the WH-based learning algorithms, represented by ReSuMe(Ponulak & Kasiński, 2010) and SPAN (Mohemmed et al., 2012), are able to train a spiking neuron to generate spikes with accurate timing, and do not need additional tricks as STDP does. The WH learning rule (Widrow & Hoff, 1960) is a special case of the gradient descent rule where the least mean square loss is applied. Ponulak & Kasiński (2010) presented a spiking analogy to the classical WH rule for spiking neuron models, and their rule can be interpreted as an STDP-like process between a presynaptic spike train and a postsynaptic error signal. However, previous WH-based methods are constrained to train a single layer SNN since it has difficulty in propagating the teaching signal to previous layers. Sporea & Grüning (2013) extend the single layer WH-based learning rule - ReSuMe (Ponulak & Kasiński, 2010) onto multi-layers networks through BP-liked error propagation, which is practical but deviates from the original intention of exploring bio-plausible mechanisms.

In this work, we propose a **bio**-plausible **lea**rning **f**ramework - BioLeaF, underpinned by two key components: **1)** a microcircuit architecture consisting of two types of spiking neurons and four types of synapses as shown in Figure 1, and **2)** the STDP-based learning rules built upon our architecture.

The architecture is inspired from the predict-coding-based algorithms (Rao & Ballard, 1999; Stefanics et al., 2014). Previous works proposed several predictive-coding-inspired microcircuit architectures to realize BP-liked learning on rate-based neurons (Bastos et al., 2012; Whittington & Bogacz, 2017; Sacramento et al., 2018), where all neurons communicate with each others through continuous currents, and no explicit temporal point processes or spiking behaviors are included. This simplified setting limits the discussion of the widely used bio-plausible learning rules defined by the spike-timing correlation like STDP rule and WH rule. Our architecture differs from them in both the neuron and the synapse models. We include the more bio-plausible spiking leaky integrate-and-fire (LIF) neuron (Gerstner & Kistler, 2002) and synapses models that transmit discontinuous spikes into our architecture.

The architecture consists of two types of spiking cells - pyramidal cells and somatostatin-expressing (SOM) cells (Petreanu et al., 2009; Larkum, 2013). Each pyramidal cell i has a paired SOM cell i_p to predict its firing activity one-on-one through the same current inputs a_j, $j = 1 \cdots N$, where N is the total number of presynaptic neurons. The prediction mismatch, also interpreted as a surprise or free energy (Friston, 2010), is transmitted through top-down connections to all presynaptic neurons' apical dendrites, and acts as their error signals. A pyramidal cell's top-down output $\tilde{a}$ is modulated by its error signals, whereas a SOM cell's top-down output a_p is not. Therefore, without knowing the signal in i's apical dendrite, SOM cell i_p's prediction can only cancel out part of i's output, which leaves the error-related signal on j's apical dendrites. The summation of all top-down signals will cancel out each others pair by pair, and leaves the total error-related signals onto j, from where the layer-by-layer error backpropagation is realized.

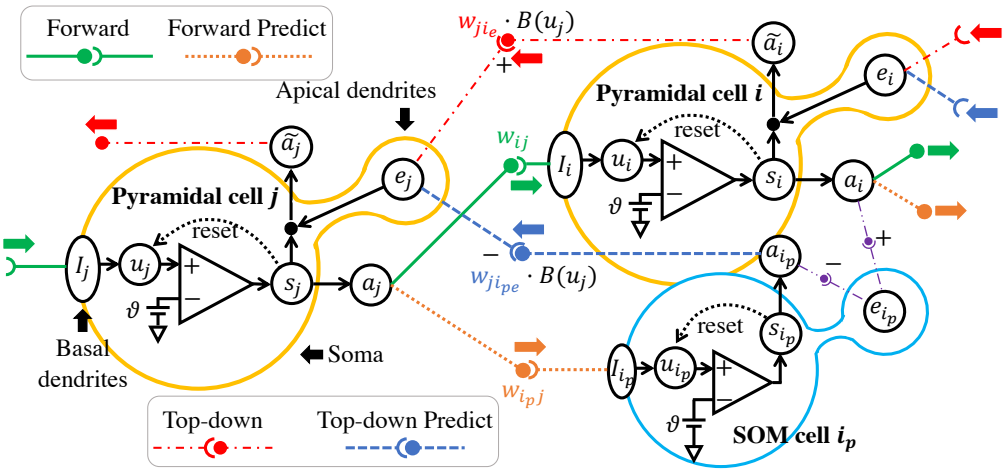


Figure 1: Our proposed Microcircuit architecture.

The learning rule built upon our microcircuit architecture is the standard STDP rule as defined in Levy & Steward (1983) with specialized choices of pre/postsynaptic signals. As comparing to typical SNNs' architectures (Shrestha & Orchard, 2018; Wu et al., 2018; Yang et al., 2021), which only have forward connected weights, we introduce three additional types of weights: forward predict, top-down, and top-down predict as in Figure 1 to support the bio-plausible learning. Therefore, to update different types of synaptic weights, our STDP updating rules need to be defined between pairs of presynaptic and postsynaptic signals locates in different components of our microcircuit architecture. A presynaptic signal is an output spike train located in the presynaptic neuron like s_j for both weights w_{ij} and $w_{i_p j}$, or s_{i_p} for $w_{ji_{pe}}$ following Levy & Steward (1983), and a postsynaptic signal is an error signal located in the postsynaptic neuron like e_i for weights w_{ij} and e_j for $w_{ji_{pe}}$.

More generally, we analytically show that the the proposed framework is equivalent to the BP-based learning rules under certain settings. To derive a BP-based learning rules which propagates continuous-valued loss signal through discontinuous all-or-none firing activity, some approximation methods are applied following previous works (Shrestha & Orchard, 2018; Wu et al., 2018; Yang et al., 2021). Yet such approximation surprisingly aligned with the standard STDP rules under our microcircuit architecture with only minor differences. We empirically build a 2-layers toy example to evaluate the learning ability of BioLeaF. Deeper than a single layer breaks the limit of the previous WH-based learning rules. Then, by benchmarking on the datasets including MNIST (LeCun, 1998) and CIFAR10 (Krizhevsky et al., 2009), the proposed BioLeaF also exhibits comparable accuracy with other BP-based methods when extended to multi-layers deep SNNs.

2 Microcircuit Architecture

2.1 Spiking Neuron Model

Both pyramidal cells and SOM cells are modeled by the leaky integrate-and-fire (LIF) neuron model, which is one of the most prevalent choices for describing dynamics of spiking neurons.

2.1.1 LIF Neuron Model

The dynamics of the neuronal membrane potential u of neuron i in layer l is described by:

$$\tau_m \frac{du_i^{(l)}(t)}{dt} = -u_i^{(l)}(t) + I_i^{(l)}(t) + \eta_i^{(l)}(t), \tag{1}$$

where $I_i^{(l)}$ is the total input of synaptic currents, and $\eta_i^{(l)}(t)$ denotes the reset function. A spiking neuron reset its membrane potential from threshold ϑ to the resting potential v_{rest} (we set $v_{\text{rest}} = 0$) each time when it fires a spike. We model $\eta_i^{(l)}(t)$ as the time convolution (*) between a reset kernel ν and the neuron's output spike train $s_i^{(l)}$: $\eta_i^{(l)}(t) = (\nu * s_i^{(l)})(t)$. The reset kernel $\nu(t) = -\vartheta\delta(t)$.

The amount of resetting is equal to the threshold ϑ (we set $\vartheta = 1$), and $\delta(t)$ is the Dirac delta function. The neuron's output spike train is also modeled by a serious of delta functions as:$s_i^{(l)} = \sum_f \delta(t - t_{i(f)}^{(l)})$. Here, $t_{i(f)}^{(l)}$ represent the firing time of the fth spike of neuron i in layer (l). An output spike is generated once the membrane potential reaches the threshold ϑ. Following Shrestha & Orchard (2018), we define the spike function as:

$$f_s(u) : u \to s, s(t) := s(t) + \delta(t - t_{(f+1)}), t_{(f+1)} = \min\left\{t : u(t) = \vartheta, t > t_{(f)}\right\}. \tag{2}$$

2.2 Synaptic Currents

We model the general total input current on neuron i as: $I_i(t) = \sum_j w_{ij} \sum_f \alpha_{ij}(t - t_{j(f)})$. The modeled total input is the weighted sum over all current pulses:

$$\alpha_{ij}\left(t - t_{j(f)}\right) = g_{ij}\left(t - t_{j(f)}\right) \cdot \left[E_{\text{syn}} - u_i(t)\right], \tag{3}$$

where E_{syn} is the reversal potential for the synaptic current. We set $E_{\text{syn}} \gg \vartheta$ in all types of synapses (Destexhe et al., 1998), so the membrane potential dependency can be neglected, and the term $[E_{\text{syn}} - u_i(t)]$ can be treated as a constant and absorbed into weights. $g_{ij}(t - t_{j(f)})$ is the synaptic conductance change. We modeled it following Eyal et al. (2018), but simplified the double exponential function to a single exponential decaying function and have:

$$g_{ij}\left(t - t_{j(f)}\right) = B_i(t) \cdot (1/\tau_s) \cdot exp\left[-\left(t - t_{j(f)}\right)/\tau_s\right], \tag{4}$$

where $B_i(t)$ is a membrane potential dependent gating function. The synapses in different location have different properties, where the two types of synapse we apply are named as the *forward-related-type (F-type)* and the *error-related-type (E-type)*. We introduce them one by one as following:

F-Type Synapses:

The connections from the outputs of pyramidal cells in one layer to the basal dendrites of pyramidal cells in their next layer carrying important feature information build up the main architecture in a spiking neural networks. When training is finished, only these forward connections are needed to realize inference. We implement these forward connections $w_{ij}^{(l)}$ together with their paired forward predict connections $w_{i_pj}^{(l)}$ with the F-type synapses.

$B_i(t)$ is set to 1 (voltage independent conductance) as a general setting for AMPA-based conductance (Eyal et al., 2018). We follow this setting and the input current is then simplified to a widely-used alpha function with time constant τ_s:

$$\alpha_{ij}\left(t - t_{j(f)}\right) = (1/\tau_s) \cdot exp\left[-\left(t - t_{j(f)}\right)/\tau_s\right]. \tag{5}$$

Under which, the synaptic current is independent from neuron i, and all postsynaptic current (PSC) generated from neuron j can be uniformly expressed by one variable a_j as:

$$a_j(t) = \sum_f \alpha_{ij}(t - t_{j(f)}) = (s_j * \epsilon)(t), \quad \epsilon(t) = (1/\tau_s) \cdot exp(-t/\tau_s)H(t), \tag{6}$$

where (*) represents the time convolution. $\epsilon(\cdot)$ is the impulse response. $H(\cdot)$ represents the Heaviside step function: $H(t) = 1, t \geq 0$ and $H(t) = 0, t < 0$.

A fully connected layer can be described through the current flows as:

$$I_i^{(l)}(t) = \sum_{j=1}^{N^{(l-1)}} w_{ij}^{(l)} \sum_f \alpha_{ij}(t - t_{j(f)}^{(l-1)}) = \sum_{j=1}^{N^{(l-1)}} w_{ij}^{(l)} a_j^{(l-1)}(t). \tag{7}$$

Other layers like convolution layers can be easily converted to fully connected layers. Similarly, the total current of SOM cell $I_{i_p}^{(l)}$ is modeled as: $I_{i_p}^{(l)}(t) = \sum_{j=1}^{N^{(l-1)}} w_{i_pj}^{(l)} a_j^{(l-1)}$, where the footnote p represent the predict-related, or the SOM-related variables.

E-Type Synapses:

All other connections are modeled by E-type synapses, which differs from the F-type synapses by their postsynaptic voltage dependent property. We model the voltage dependent gating function

$B_i(t)$ like how prior works model the NMDA-based synapses (Eyal et al., 2018). $B_i(t)$ has a shape that peaks when the postsynaptic cell's membrane potential $u_i(t)$ reaches the threshold ϑ, and decrease as $u_i(t)$ moves away from ϑ. Such a shape acts as surrogate derivative function when compared to BP-based methods, which will be fully discussed in the following section.

We define $B_i(t)$ of our synapses on the apical dendrites as following:

$$B_i(t) = \frac{g_{\max}}{1 + exp\left[-k\left(u_i(t) - u_0\right)\right] \cdot [\mathrm{Mg}^{2+}] \cdot n}, \tag{8}$$

where the extracellular magnesium concentration $[\mathrm{Mg}^{2+}]$ was 1 mM in the model. We shift the voltage dependency of $B_i(t)$ by $u_0 = \vartheta$, and tune the parameters $g_{\max}$, k and n to fits the function B into our simplified setting where $v_{\mathrm{rest}} = 0$, and $\vartheta = 1$.

The error backpropagation is achieved by the corporation between pairs of pyramidal cells and SOM cells. Higher level pyramidal cells' top-down output currents are coupled by the error signals located in their apical dendrites. We model this coupling effect by the current sum of both a neuron's PSC and its error signal as: $\widetilde{a}_i^{(l)}(t) = a_i^{(l)}(t) + e_i^{(l)}(t)$. A pyramidal cell's top-down connection contributes positively with weights $w_{ji_e}^{(l)}$ onto previous layers' pyramidal cells' apical dendrites, and its paired SOM cell contribute negatively with weights $w_{ji_{pe}}^{(l)}$, where the footnotes p stands for SOM-related, and the footnotes e stands for error-related. We express the total error signals on the pyramidal cell j's apical dendrites as:

$$e_j^{(l-1)}(t) = B_j^{(l-1)}(t) \left[\sum_{i=1}^{N^{(l)}} \left(w_{ji_e}^{(l)} \widetilde{a}_i^{(l)}(t) - w_{ji_{pe}}^{(l)} a_{i_p}^{(l)}(t) \right) \right], \tag{9}$$

For the output layer, the apical dendrites receives the one-on-one error signal from higher brain areas to realize supervised learning.

$$e_i^{(l_N)}(t) = B_i^{(l_N)}(t) \left[a_i^{\mathrm{target}}(t) - a_i^{(l_N)}(t) \right] \tag{10}$$

In our framework, SOM cells mimic the behavior of the same layer's pyramidal cells one-on-one. Therefore, a one-to-one nudging signal from a pyramidal cell to its corresponding SOM cell (as the dashed purple connections in Figure 1) are needed. Together with the negative feedback output currents that SOM cells generated themselves, we get the local error signals $e_{i_p}^{(l)}$ of an SOM cell i in layer (l):

$$e_{i_p}^{(l)}(t) = \alpha_{i_p j}^{(l)}(t) - \alpha_{i_p i_p}^{(l)}(t) = B_{i_p}^{(l)}(t) \left[a_i^{(l)}(t) - a_{i_p}^{(l)}(t) \right]. \tag{11}$$

Although this simplified one-to-one setting impose special constrains on the neural network's connectivity, the recent monosynaptic experiments confirm that the SOM cells do receive top-down connections which may encode such teaching information (Leinweber et al., 2017).

The SOM cells differs from the pyramidal cells by the sign of their output currents, but it does not mean that we fix the type of a cell to be excitatory or inhibitory. Instead, we allow the synapses' connection weights to move across zero freely and change the sign of its current, which is a general setting in previous works (Ponulak & Kasiński, 2010; Sacramento et al., 2018). When currents from both SOM cells and pyramidal cells summed together into the SOM cells or the apical dendrites as in (11) and (9), a minus sign is added for SOM cell's outputs.

3 Bio-plausible Learning Rules

3.1 Spike-Timing-Dependent-Plasticity (STDP)

We first introduce a popular version of the STDP rules: $\Delta w_{ij} = \sum_m \sum_n \kappa_{\mathrm{STDP}}(t_i^m - t_j^n)$, where κ_{STDP} is the STDP kernel function, which is modeled by the two-sides exponential decaying function defined as:

$$\kappa_{\mathrm{STDP}}(\Delta t) = \begin{cases} A^+ \cdot exp(-\Delta t/\tau_+), & \Delta t > 0 \\ -A^- \cdot exp(\Delta t/\tau_-), & \Delta t < 0 \end{cases} \tag{12}$$

Meantime, we define a reversed STDP kernel function $\tilde{\kappa}_{\mathrm{STDP}}(t) = \kappa_{\mathrm{STDP}}(-t)$. Recalling that all spike trains are a serious of delta function $s = \sum_f \delta(t - t_{(f)})$, and considering the delta function's sampling property $\int [f(t)\delta(t-T)]\, dt = f(T)$, we rewrite the STDP updating rule into two equivalent forms:

$$\Delta w_{ij} = \eta_{ij} \int (s_j * \kappa_{\mathrm{STDP}})(t) \cdot s_i(t)\, dt = \eta_{ij} \int s_j(t) \cdot (s_i * \tilde{\kappa}_{\mathrm{STDP}})(t)\, dt, \tag{13}$$

where η_{ij} is the learning rate. Although η_{ij} can be absorbed into A^+ and A^-, we explicitly define it for clarity.

Since the WH rule can be interpreted as an STDP-like process between the presynaptic spike trains s_j and an error signal $e_i := s_i^{\mathrm{teach}} - s_i$. In this work, we represent both the STDP rule and the WH rule uniformly as $\mathrm{STDP}(\cdot, \cdot)$, where the STDP rule is: $\mathrm{STDP}(s_{pre}, s_{post})$, and the WH rule is: $\mathrm{STDP}(s_{pre}, e_{post})$. In addition, WH rule's kernel function κ_{WH} is usually equal to κ_{STDP} as in (12). Defining the reverse kernel function $\tilde{\kappa}_{\mathrm{WH}}(t) = \kappa_{\mathrm{WH}}(-t)$, we have the weight updating rule of the WH rule as:

$$\Delta w_{ij} = \eta_{ij} \int (s_j * \kappa_{\mathrm{WH}})(t) \cdot e_i(t)\, dt = \eta_{ij} \int s_j(t) \cdot (e_i * \tilde{\kappa}_{\mathrm{WH}})(t)\, dt. \tag{14}$$

It is noteworthy that the WH-based rules provides a fixed point in the weight space, $\Delta w_{ij} = 0$ when $e_i = 0$, which means $s_i^{\mathrm{teach}} = s_i$. It is a global positive attractor under certain conditions (Ponulak & Kasiński, 2010).

3.2 Our bio-plausible learning rules

We first conclude our synaptic learning rules as following:

$$\Delta w_{ij}^{(l)} = \mathrm{STDP}(s_j^{(l-1)}, e_i^{(l)}) = \eta_{ij} \int \left(s_j^{(l-1)}(t) * \kappa_{\mathrm{STDP}} \right)(t) \cdot e_i^{(l)}(t) dt \tag{15}$$

$$\Delta w_{i_p j}^{(l)} = \mathrm{STDP}(s_j^{(l-1)}, e_{i_p}^{(l)}) = \eta_{i_p j} \int \left(s_j^{(l-1)}(t) * \kappa_{\mathrm{STDP}} \right)(t) \cdot e_{i_p}^{(l)}(t) dt \tag{16}$$

$$\Delta w_{j i_{pe}}^{(l)} = \mathrm{STDP}(s_{i_p}^{(l)}, e_j^{(l-1)}) = \eta_{j i_{pe}} \int \left(s_{i_p}^{(l)}(t) * \kappa_{\mathrm{STDP}} \right)(t) \cdot e_j^{(l-1)}(t) dt \tag{17}$$

where $\eta_{ij}, \eta_{i_p j}$, and $\eta_{j i_{pe}}$ are three different learning rates for these three types of synaptic connections. In the above three equations, all the adjusting rules of a synapse can be described as a STDP process between a presynaptic current and a postsynaptic error signal, which exploits great biological plausibility. The top-down synaptic weights $w_{j i_e}^{(l)}$ are set to be equal to the bottom-up forward weights $w_{ij}^{(l)}$ for simplicity. One can also try to fixed the top-down weights $w_{j i_e}^{(l)}$ as randomly initialized values following the idea of the feedback alignment (Sacramento et al., 2018).

We interpret each type of synapses' learning rule as following:

In (15), $w_{ij}^{(l)}$ represents the forward weights. It's fixed point $e_i^{(l)}(t) = B_i^{(l_N)}(t)[a_i^{\mathrm{target}}(t) - a_i^{(l_N)}(t)] = 0$ forms a positive attractor to minimize the error signal. When training is converged, we expect $a_i^{(l_N)} \approx a_i^{\mathrm{target}}$ for all pyramidal cells $(i = 1, \ldots, N^{(l)}), (l = 1, \ldots, l_N)$.

In (16), $w_{i_p j}^{(l)}$ stands for the predictive connections. Its fixed point in our learning rule is $e_{i_p}^{(l)}(t) = B_{i_p}^{(l)}(t)[a_i^{(l)}(t) - a_{i_p}^{(l)}(t)] = 0$, which means $a_i^{(l)} = a_{i_p}^{(l)}$, and $s_i^{(l)} = s_{i_p}^{(l)}$. As compared to (15), the learning rule of (16) does not have a fixed target signal, but needs to follow the continuously changing behavior of each SOM cell's corresponding pyramidal cell during training.

The learning rule (17) minimizes the norm of $e_j^{(l-1)}(t)$ in (9) through adjusting $w_{j i_{pe}}^{(l)}$, which is equivalent to solving a linear equation: $\sum_{i=1}^{N^{(l)}} w_{j i_{pe}}^{(l)} a_{i_p}^{(l)}(t) = \sum_{i=1}^{N^{(l)}} w_{j i_e}^{(l)} (a_i^{(l)}(t) + e_i^{(l)}(t))$.

Considering the Current-Time as a continuous $f(X) \leftrightarrow X$ function space, where all SOM cells' PSCs $a_{i_p}^{(l)}, (i = 1, \ldots N_l)$ form a basis of the space. Then the goal here is to restore the summed

currents of all pyramidal cells in layer (l) through these basis functions. When a predictive weight $w_{i_p j}^{(l)}$ is well adjusted, $a_i^{(l)}$ approximately equals to $a_{i_p}^{(l)}$, then an obvious solution to the equation above will be letting $w_{ji_{pe}}^{(l)}$ equal to $w_{ji_e}^{(l)}$, if we consider the error currents $e_i^{(l)}(t)$ as the orthogonal uncorrelated signal to the basis.

4 Relationship to the BP Algorithm

4.1 Backpropagation Flow

Since the SOM cells are auxiliary in our architecture, we only introduce how backpropagation works in a general SNN without SOM cells.

Consider a general loss function $L = \int_t E(t)dt$, which is defined on the output layers' PSC $a_i^{(l_N)}(t),\ i = 1, \ldots, N^{(l)}$, where l_N is the total number of layers. The differentiable property of a loss function requires that $\partial L/\partial a^{(l_N)}(t)$ exist. We name the partial derivative on the layer (l)'s PSC as:$d_i^{(l)}(t) := \frac{\partial L}{\partial a_i^{(l)}(t)}$.

Taking the $l2$ distance between two PSCs, or the van Rossum distance (van Rossum, 2001) between two spike trains, as an example:

$$L = \frac{1}{2}\int \left[\left(a_i^{\text{target}} - a_i^{(l_N)}\right)(t)\right]^2 dt = \frac{1}{2}\int \left[\left((s_i^{\text{target}} * \epsilon) - (s_i^{(l_N)} * \epsilon)\right)(t)\right]^2 dt \tag{18}$$

When computing the gradient of loss with respect to the synaptic weight of the last layer $w_{ij}^{(l_N)}$: $\partial L/\partial w_{ij}^{(l_N)} = \int d_i^{(l_N)}(t)\left(\epsilon * \partial s_i^{(l_N)}/\partial w_{ij}^{(l_N)}\right)(t)dt$, one may find the derivative $\partial s_i^{(l_N)}/\partial w_{ij}^{(l_N)}$ is ill-defined due to spiking neurons' discontinuous all-or-none firing activities. Following (Zenke & Ganguli, 2018), we substitute this term by: $(\sigma'(u_i^{(l_N)}) \cdot \partial u_i^{(l_N)}/\partial w_{ij}^{(l_N)})$, and further approximate the term $(\partial u_i^{(l_N)}/\partial w_{ij}^{(l_N)}) \approx a_j^{(l_N-1)}$ by omitting the temporal dependency of membrane potentials. We have the weights updating rule as:

$$\begin{aligned}
\Delta w_{ij}^{(l_N)} &= -\eta \frac{\partial L}{\partial w_{ij}^{(l_N)}} = \eta \int -d_i^{(l_N)}(t)\left(\epsilon * (\sigma'(u_i^{(l_N)}) \cdot a_j^{(l_N-1)})\right)(t)dt \\
&\approx \eta \int \underbrace{\left(-d_i^{(l_N)}\sigma'(u_i^{(l_N)})\right)(t)}_{\text{post}} \cdot \underbrace{\left(\epsilon * (s_j^{(l_N-1)} * \epsilon))\right)(t)}_{\text{pre}} = \eta \int e_i^{(l_N)}(t) \cdot \left(s_j^{(l_N-1)} * \kappa_{\text{BP}}\right)(t),
\end{aligned} \tag{19}$$

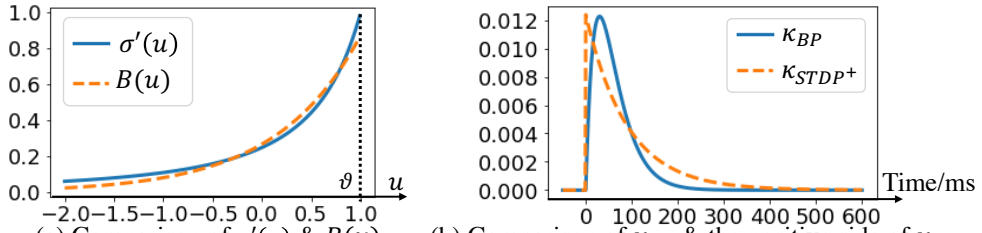
where η is the learning rate, $e_i^{(l_N)} := -d_i^{(l_N)}\sigma'(u_i^{(l_N)})$, and the kernel function of BP algorithm $\kappa_{\text{BP}}(t) := (\epsilon * \epsilon)(t) = \left[(t/\tau_s^2) \cdot exp(-t/\tau_s) \cdot H(t)\right]$. In the second line, we approximately switch the order of time convolution and product to separate variables of presynaptic neurons and postsynaptic neurons. Then to further propagate the gradient to previous layers, we calculate the partial derivative of loss with respect to hidden layer's PSCs as:

$$d_j^{(l-1)}(t) = \sum_{i=1}^{N^{(l)}} \left(d_i^{(l)} * \epsilon\right)(t) \cdot \sigma'\left(u_i^{(l)}(t)\right) w_{ij}^{(l)} \approx \sum_{i=1}^{N^{(l)}} w_{ij}^{(l)}\left(d_i^{(l)} \cdot \sigma'(u_i^{(l)})\right)(t) = -\sum_{i=1}^{N^{(l)}} w_{ij}^{(l)} e_i^{(l)}, \tag{20}$$

Here we omit the temporal dependency $(*\epsilon)$ for simplicity. Since BP is not our contribution, we discuss different omitting methods of all previous works and the detailed derivatives including the full dependency in the appendix. With $d_j^{(l-1)}(t)$ calculated, an hidden layer follows the same rule to propagate from $d_j^{(l-1)}$ to $(\partial L/\partial w_{jm}^{(l-1)})$ as the output layer.

4.2 Comparing BP with our framework

We analyses our bio-plausible learning rule from the optimization point of view.



(a) Comparison of $\sigma'(u)$ & $B(u)$ (b) Comparison of κ_{BP} & the positive side of κ_{STDP}

Figure 2: (a) $\sigma'(u) = 1/(1+|u-1|)^2)$ is defined following Zenke & Ganguli (2018), and the parameters of $B(u)$ used here are: $u_0 = \vartheta = 1$, $g_{\max} = 109.45$, $k = 1.18$ and $n = 124.33$. (b) $\kappa_{\text{BP}} = \epsilon * \epsilon$, with $\tau_s = 30$ ms. The positive side of $\kappa_{\text{STDP}} = A^+ exp(-t/\tau_+)H(t)$ has parameters: $A^+ = 0.0124$, and $\tau_+ = 89.73$ ms.

In the output layer, the error signal in (10) corresponds to $\left(-d_i^{(l_N)}\sigma'(u_i^{(l_N)})\right)$, which is the postsynaptic part in (19). The term $-d_i^{(l_N)}$ corresponds to $[a_i^{\text{target}}(t) - a_i^{(l_N)}(t)]$ in (10), which implies that the equivalent loss function we apply for the bio-plausible rule is also the van Rossum distance. For other loss functions, one can safely substitute $[a_i^{\text{target}}(t) - a_i^{(l_N)}(t)]$ with their own $(-d_i^{(l_N)})$. The auxiliary $\sigma'(u_i^{(l_N)})$ function corresponds to the voltage dependent gating function $B_i^{(l_N)}$. One may find that with proper parametrization they can almost overlap as shown in Figure 2 (a). Both of them reshapes the error signal depending on the neuron's membrane potential, where the gradients farther from the threshold are weakened.

With error signals clearly defined, we find both the BP-based learning rule (19) and the previously described bio-plausible STDP-based learning rule (15) follow a surprisingly similar form:

$$\Delta w_{ij} \propto \int (s_{\text{pre}} * \kappa) \cdot e_{\text{post}}. \tag{21}$$

As shown in Figure 2 (b), the shape of BP's kernel function κ_{BP} is highly similar to the positive side of the STDP kernel κ_{STDP}, where the value peaks near zero, and decays gradually. Such equivalence gives theoretical analysis of what should the kernel looks like from the optimization point of view and provides possibly explanations of why the negative side of STDP learning rule usually dampen the performance, and usually been omitted in previous works to boost the performance (Ponulak & Kasiński, 2010). In the experiment part, we also ignored the negative side of STDP kernel.

When further propagating the gradient to previous layers, the more complex predictive-coding-inspired architectures are involved. As in Sacramento et al. (2018), we also name the ideal state where $w_{ij}^{(l)} = w_{i_{pj}}^{(l)} = w_{ji_{pe}}^{(l)} = w_{ji_e}^{(l)}$ as *self-predicting* (*self-predicting* is needed theoretically, but not experimentally as shown in the next section). Under which, the summed error signal on the apical dendrites of a pyramidal cell in (9) are simplified to:

$$e_j^{(l-1)}(t) = B_j^{(l-1)}(t) \sum_{i=1}^{N^{(l)}} \left(w_{ij}^{(l)}(a_i^{(l)}(t) + e_i^{(l)}(t)) - w_{ij}^{(l)} a_i^{(l)}(t) \right) = B_j^{(l-1)}(t) \sum_{i=1}^{N^{(l)}} w_{ij}^{(l)} e_i^{(l)}(t). \tag{22}$$

Comparing to (20), which propagates the negative weighted sum of $e_i^{(l)}$ to $d_j^{(l-1)}$, and further calculate $e_j^{(l-1)} = -d_j^{(l-1)}\sigma'(u_j^{(l)})$, our rule yields more symmetry, where the error signals e flow through layers without any intermediate variables. These two rules are equivalent when pairing $B(u)$ to $\sigma'(u)$ as in Figure 2 (a).

We conclude their similarity and difference as following: 1) Both of the learning rules share a same form: $\Delta w_{ij} \propto \int (s_{\text{pre}} * \kappa) \cdot e_{\text{post}}$. 2) The surrogate derivatives $\sigma'(u_i^{(l)})$ in BP is correspondingly achieved by $B_i^{(l)}$ of the E-type synapses in our framework. 3) The shape of BP's kernel function κ_{BP} is similar to the positive side of κ_{STDP}. 4) For both rules, the error backpropagation between two layers are equivalent when our network in its *self-predicting* state. And the one difference these two methods have is: The kernel function κ_{BP} only has the positive side ($t > 0$), but κ_{STDP} has double sides.

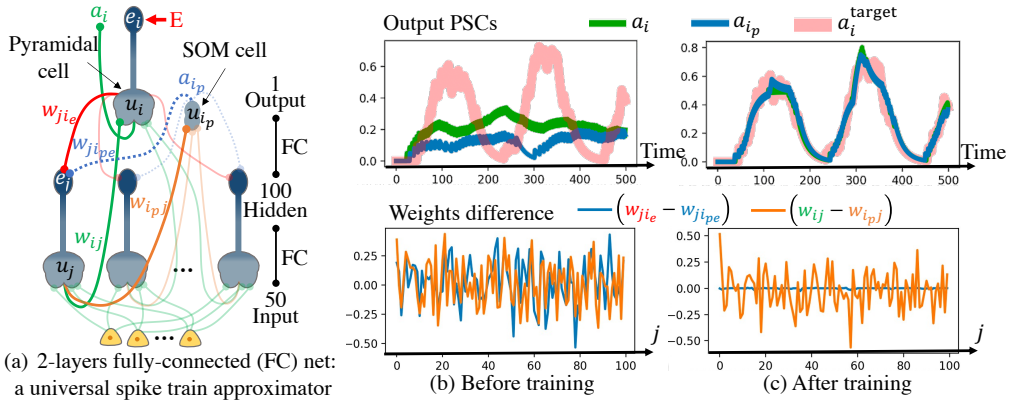


Figure 3: Universal spike train approximator experiment

5 Experimental Results

5.1 Universal Spike Train Approximator

To test the learning ability of our proposed bio-plausible learning rules, we build up a 2-layers SNN to fit a random target spike train (PSC) from randomly generated inputs as shown in Figure 3 (a). One SOM cell paired with the output pyramidal cell helps it propagate error backwards and update the input weights. More detail settings can be found in the appendix. In Figure 3 (b) and (c), the upper two sub-figures exploit that both the output a_i and the predict output a_{i_p} are able to fit our randomly assigned sinusoidal target PSC a_i^{target}. The lower two sub-figures shows the difference between paired weights, where the two backward weights $w_{ji_{pe}}^{(l)} \approx w_{ji_e}^{(l)}$ after training, yet the other pair remains different. Such difference would not hinder the training because the success of error backpropagation only requires $a_i \approx a_{i_p}$.

5.2 The Results on the MNIST Dataset and the CIFAR-10 Dataset

The proposed framework is compared with other BP-based rules on two widely used real-world datasets: MNIST (LeCun, 1998) and CIFAR-10 (Krizhevsky et al., 2009). Previous works usually use the fixed-step first-order forward Euler method to discretize continuous membrane voltage updates over a set of discrete time steps, we also following this setting and take several measures to guarantee a fair comparison: 1) Mirroring $B(u)$ with ϑ to get $B(u)$'s value when $u > \vartheta$. 2) Setting $\kappa_{\text{STDP}}[t] = 1$ when $t = 0$, and $\kappa_{\text{STDP}}[t] = 0$ when $t \neq 0$. 3) All the comparisons are made under a SNN's self-predicting state. The results are concluded in table 1. Our method gains comparable performance as compared to BP-based works.

Table 1: Performances comparison of different methods on the MNIST and CIFAR10 datasets

	MNIST		CIFAR10	
Method	#Steps	BestAcc	#Steps	BestAcc
SLAYER (Shrestha & Orchard, 2018)	300	99.41%	null	null
TSSL-BP (Zhang & Li, 2020)	5	99.53%	5	89.22%
NA (Yang et al., 2021)	5	99.69%	5	91.76%
This work	**5**	**99.46%**	**5**	**86.88%**

MNIST SNN structure: 15C5-P2-40C5-P2-300
CIFAR10 SNN structure: 96C3-256C3-P2-384C3-P2-384C3-256C3-1024-1024

6 Conclusion

We proposed a new bio-plausible learning framework, BioLeaF, consisting of two key components: an architecture, and its paired learning rules. BioLeaF leverages previous bio-plausible works' limitation, and bridges the gap between the bio-plausible approach and the BP-based approach both analytically and experimentally. The equivalence of these two approaches are demonstrated under a special setting, and the comparable experimental performance of them are benchmarked on MNIST and CIFAR10 datasets. This work may provide new insights on both approaches.

REFERENCES

Andre M Bastos, W Martin Usrey, Rick A Adams, George R Mangun, Pascal Fries, and Karl J Friston. Canonical microcircuits for predictive coding. *Neuron*, 76(4):695–711, 2012.

Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards AI. In *Large Scale Kernel Machines*. MIT Press, 2007.

Mike Davies, Narayan Srinivasa, Tsung-Han Lin, Gautham Chinya, Yongqiang Cao, Sri Harsha Choday, Georgios Dimou, Prasad Joshi, Nabil Imam, Shweta Jain, et al. Loihi: A neuromorphic manycore processor with on-chip learning. *Ieee Micro*, 38(1):82–99, 2018.

Mike Davies, Andreas Wild, Garrick Orchard, Yulia Sandamirskaya, Gabriel A Fonseca Guerra, Prasad Joshi, Philipp Plank, and Sumedh R Risbud. Advancing neuromorphic computing with loihi: A survey of results and outlook. *Proceedings of the IEEE*, 109(5):911–934, 2021.

Sophie Denève, Alireza Alemi, and Ralph Bourdoukan. The brain as an efficient and robust adaptive learner. *Neuron*, 94(5):969–977, 2017.

Alain Destexhe, Zachary F Mainen, Terrence J Sejnowski, et al. Kinetic models of synaptic transmission. *Methods in neuronal modeling*, 2:1–25, 1998.

Peter U Diehl and Matthew Cook. Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in computational neuroscience*, 9:99, 2015.

Guy Eyal, Matthijs B Verhoog, Guilherme Testa-Silva, Yair Deitcher, Ruth Benavides-Piccione, Javier DeFelipe, Christiaan PJ De Kock, Huibert D Mansvelder, and Idan Segev. Human cortical pyramidal neurons: from spines to spikes via models. *Frontiers in cellular neuroscience*, 12:181, 2018.

Paul Ferré, Franck Mamalet, and Simon J Thorpe. Unsupervised feature learning with winner-takes-all based stdp. *Frontiers in computational neuroscience*, 12:24, 2018.

Karl Friston. The free-energy principle: a unified brain theory? *Nature reviews neuroscience*, 11 (2):127–138, 2010.

Wulfram Gerstner and Werner M Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. *Deep learning*, volume 1. MIT Press, 2016.

Yunzhe Hao, Xuhui Huang, Meng Dong, and Bo Xu. A biologically plausible supervised learning method for spiking neural networks using the symmetric stdp rule. *Neural Networks*, 121:387–395, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Donald Olding Hebb. *The organization of behavior: A neuropsychological theory*. Psychology Press, 1949.

Geoffrey E. Hinton, Simon Osindero, and Yee Whye Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006.

Bernd Illing, Wulfram Gerstner, and Johanni Brea. Biologically plausible deep learning—but how far can we go with shallow networks? *Neural Networks*, 118:90–101, 2019.

Saeed Reza Kheradpisheh, Mohammad Ganjtabesh, Simon J Thorpe, and Timothée Masquelier. Stdp-based spiking deep convolutional neural networks for object recognition. *Neural Networks*, 99:56–67, 2018.

Jinseok Kim, Kyungsu Kim, and Jae-Joon Kim. Unifying activation-and timing-based learning rules for spiking neural networks. *Advances in Neural Information Processing Systems*, 2020a.

Seijoon Kim, Seongsik Park, Byunggook Na, and Sungroh Yoon. Spiking-yolo: Spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11270–11277, 2020b.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Matthew Larkum. A cellular mechanism for cortical associations: an organizing principle for the cerebral cortex. *Trends in neurosciences*, 36(3):141–151, 2013.

Yann LeCun. The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*, 1998.

Chankyu Lee, Priyadarshini Panda, Gopalakrishnan Srinivasan, and Kaushik Roy. Training deep spiking convolutional neural networks with stdp-based unsupervised pre-training followed by supervised fine-tuning. *Frontiers in neuroscience*, 12:435, 2018.

Marcus Leinweber, Daniel R Ward, Jan M Sobczak, Alexander Attinger, and Georg B Keller. A sensorimotor circuit in mouse cortex for visual flow predictions. *Neuron*, 95(6):1420–1432, 2017.

WB Levy and O Steward. Temporal contiguity requirements for long-term associative potentiation/depression in the hippocampus. *Neuroscience*, 8(4):791–797, 1983.

Jesus L Lobo, Javier Del Ser, Albert Bifet, and Nikola Kasabov. Spiking neural networks and online learning: An overview and perspectives. *Neural Networks*, 121:88–100, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Ammar Mohemmed, Stefan Schliebs, Satoshi Matsuda, and Nikola Kasabov. Span: Spike pattern association neuron for learning spatio-temporal spike patterns. *International journal of neural systems*, 22(04):1250012, 2012.

Martin M Monti and Daniel N Osherson. Logic, language and the brain. *Brain research*, 1428: 33–42, 2012.

Bernhard Nessler, Michael Pfeiffer, Lars Buesing, and Wolfgang Maass. Bayesian computation emerges in generic cortical microcircuits through spike-timing-dependent plasticity. *PLoS computational biology*, 9(4):e1003037, 2013.

Priyadarshini Panda and Kaushik Roy. Unsupervised regenerative learning of hierarchical features in spiking deep networks for object recognition. In *2016 International Joint Conference on Neural Networks (IJCNN)*, pp. 299–306. IEEE, 2016.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Leopoldo Petreanu, Tianyi Mao, Scott M Sternson, and Karel Svoboda. The subcellular organization of neocortical excitatory connections. *Nature*, 457(7233):1142–1145, 2009.

Filip Ponulak and Andrzej Kasiński. Supervised learning in spiking neural networks with resume: sequence learning, classification, and spike shifting. *Neural computation*, 22(2):467–510, 2010.

Ximing Qiao, Yukun Yang, and Hai Li. Defending neural backdoors via generative distribution modeling. *Advances in Neural Information Processing Systems*, 2019.

Marcus E Raichle, Ann Mary MacLeod, Abraham Z Snyder, William J Powers, Debra A Gusnard, and Gordon L Shulman. A default mode of brain function. *Proceedings of the National Academy of Sciences*, 98(2):676–682, 2001.

Rajesh PN Rao and Dana H Ballard. Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects. *Nature neuroscience*, 2(1):79–87, 1999.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

João Sacramento, Rui Ponte Costa, Yoshua Bengio, and Walter Senn. Dendritic cortical microcircuits approximate the backpropagation algorithm. *Advances in Neural Information Processing Systems*, 2018.

Daniel J Saunders, Hava T Siegelmann, Robert Kozma, et al. Stdp learning of image patches with convolutional spiking neural networks. In *2018 international joint conference on neural networks (IJCNN)*, pp. 1–7. IEEE, 2018.

Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.

Sumit Bam Shrestha and Garrick Orchard. SLAYER: spike layer error reassignment in time. In *Advances in Neural Information Processing Systems*, pp. 1419–1428, 2018.

Ioana Sporea and André Grüning. Supervised learning in multilayer spiking neural networks. *Neural computation*, 25(2):473–509, 2013.

Gábor Stefanics, Jan Kremláček, and István Czigler. Visual mismatch negativity: a predictive coding view. *Frontiers in human neuroscience*, 8:666, 2014.

MM Taylor et al. The problem of stimulus structure in the behavioural theory of perception. *South African Journal of Psychology*, 3:23–45, 1973.

Mark CW van Rossum. A novel spike distance. *Neural computation*, 13(4):751–763, 2001.

John Von Neumann. *The computer and the brain*. Yale University Press, 2012.

James CR Whittington and Rafal Bogacz. An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity. *Neural computation*, 29(5): 1229–1262, 2017.

Bernard Widrow and Marcian E Hoff. Adaptive switching circuits. Technical report, Stanford Univ Ca Stanford Electronics Labs, 1960.

Yujie Wu, Lei Deng, Guoqi Li, Jun Zhu, and Luping Shi. Spatio-temporal backpropagation for training high-performance spiking neural networks. *Frontiers in neuroscience*, 12:331, 2018.

Yukun Yang, Wenrui Zhang, and Peng Li. Backpropagated neighborhood aggregation for accurate training of spiking neural networks. In *International Conference on Machine Learning*, pp. 11852–11862. PMLR, 2021.

Friedemann Zenke and Surya Ganguli. Superspike: Supervised learning in multilayer spiking neural networks. *Neural computation*, 30(6):1514–1541, 2018.

Wenrui Zhang and Peng Li. Temporal spike sequence learning via backpropagation for deep spiking neural networks. *Advances in Neural Information Processing Systems*, 33, 2020.