

Towards biologically plausible learning in neural networks

Citation for published version (APA):

García Fernández, J., Hortal, E., & Mehrkanoon, S. (2021). Towards biologically plausible learning in neural networks. In *2021 IEEE Symposium Series on Computational Intelligence (SSCI)* (pp. 1-8). [9659539] IEEE Canada. <https://doi.org/10.1109/SSCI50451.2021.9659539>

Document status and date:

Published: 07/12/2021

DOI:

[10.1109/SSCI50451.2021.9659539](https://doi.org/10.1109/SSCI50451.2021.9659539)

Document Version:

Publisher's PDF, also known as Version of record

Document license:

Taverne

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.umlib.nl/taverne-license

Take down policy

If you believe that this document breaches copyright please contact us at:

repository@maastrichtuniversity.nl

providing details and we will investigate your claim.

Towards biologically plausible learning in neural networks

Jesús García Fernández, Enrique Hortal, Siamak Mehrkanoon

Department of Data Science and Knowledge Engineering

Maastricht University

Maastricht, The Netherlands

Abstract—Artificial neural networks are inspired by information processing performed by neural circuits in biology. While existing models are sufficient to solve many real-world tasks, they are far from reaching the potential of biological neural networks. These models are oversimplifications of their biological counterparts, omitting key features such as the spiking nature of their units or the locality during learning, among others. In this work, we, first, provide a short review of the most recent theories on biologically plausible learning and learning in Spiking Neural Networks. Then, aiming to give a step towards brain-inspired deep learning, we introduce a novel biologically plausible learning method. This approach achieves learning using only local information to each synapse, spiking units and unidirectional synaptic connections. We also propose a local solution to address the credit assignment problem based on target propagation. Finally, we evaluate our approach over three different tasks, i.e. boolean problems, image autoencoding and handwritten digit recognition.

I. INTRODUCTION

Artificial Neural Networks (ANNs) have been successfully applied to a variety of real-world problems. On some occasions, surpassing human capability, and achieving state-of-the-art in many tasks [1]–[3]. However, conventional ANNs differ from biological neural networks (BNNs) in many aspects [4]–[6]. In terms of information processing, ANNs operate based on different principles. Biological neurons communicate through electrical impulses (spikes), unlike ANNs that use continued values. Besides, the membrane of a biological neuron has an electric charge, which is modified by incoming synapses from other neurons through its dendrites. Another feature is the use of time to encode information. The spikes distributed across time is the underlying mechanism used by neurons to convey information. In terms of functionality, the differences are even more notable. While ANNs generally need a tremendous amount of data to learn, BNNs can learn a new task after witnessing a few examples. In the same way, ANNs need to be given a clear solution in order to be able to solve a specific task. That can also be seen as low efficiency when it comes to unsupervised learning, something in which BNNs are extremely good. Lastly, ANNs are highly domain-dependent. Current models generally operate to solve one specific task, which they learn using a set of specific inputs and outputs. Spiking Neural Networks (SNNs) are more realistic models of natural neural networks. They are inspired by BNNs. Yet, they are still in their early days. As BNNs, SNNs contain

spiking neurons that communicate through spikes, encoding the information over time. Furthermore, due to the non-differentiability of spikes generation, they cannot be trained using the same mechanisms used in ANNs. Although backpropagation cannot be directly applied, modifications of this algorithm can be used [7], [8]. Though their performances are not comparable to ANNs trained with backpropagation. Plus, the backpropagation algorithm has not yet been proven biologically plausible [9]. While it requires the neurons to know the activity of many other distant neurons, in biology, the synaptic weights updates depend only on the presynaptic and postsynaptic neurons [10]. On the other hand, studies have found that ANNs trained with backpropagation adopt similar representations to BNNs [11]. Hence, researchers are currently working towards approximating the backpropagation algorithm with simple local plasticity rules.

Supporting our claims in previous works [9], [12], [13], we identify the following elements as necessary for biological plausibility in artificial neural networks:

- 1) Realistic models of neurons, i.e. spiking neurons. Authors in [14] provide a comparison between different models.
- 2) Realistic synapses. It implies non-symmetric weights. Forward and feedback connections are distinct.
- 3) Local plasticity and computations. The weight modifications depend only on the presynaptic and postsynaptic neurons' activity.
- 4) Local error representation. Authors in [12] revise various recent works where this has been tackled.

Also, most of the proposed theories on learning rely on a two-phase process employing different learning rules in each phase (e.g. feedforward/feedback, positive/negative, prediction/reward). Various studies show that neuromodulatory signals, combined with feedback connection, can drive learning, resulting in different learning rules [15]. Thus, in the brain, neuromodulators such as dopamine could arguably enable different plasticity rules that originate distinct phases.

In this work, we build on the ideas presented above to propose a biologically plausible supervised learning approach. Here, we use spiking models of neurons and unidirectional non-symmetrical weights to implement the synapses. We also combine rate and temporal coding, as it occurs in biology [16]. The synaptic weights modifications are performed through

Hebbian-based learning rules, achieving local plasticity and computations. Lastly, we address the credit assignment problem with a solution inspired by target propagation [17], [18] and feedback alignment [19]. In this way, the network can also keep a local error representation. All these points are further developed in Section III. In Section II, we will review the most relevant works on biologically plausible learning in neural networks and learning in SNN, making a distinction among different categories. Due to the large number of existing approaches, we will only cover the most relevant and recent ones. Finally, Section V includes conclusions of this work and potential future research lines.

II. REVIEW OF THEORIES IN BIOLOGICALLY PLAUSIBLE LEARNING

A. *Biologically plausible learning mechanism but implausible neurons model*

Most of the ideas proposed here attempt to explain how learning occurs in the brain. They introduce learning mechanisms that can be feasible in the brain, though the chosen neuron model is oversimplified. Contrastive learning [20] and Generalized Recirculation (GeneRec) [21] are some of the first attempts. They are very similar and operate with Hebbian-based rules. In them, the weights change is proportional to a difference in neural activity over time and contains two phases. A first update is performed when the target is not provided (anti-Hebbian plasticity), and a second update is performed when the target is provided (Hebbian plasticity). Differently, [19] proposes a variation of backpropagation where the synaptic connections are not symmetric. In this variation, the feedback weights are initialized randomly and never modified, which results in sending random feedback to the presynaptic neurons. Its performance is comparable to a network trained with backpropagation. [22] proposes a method to store the errors locally over the network and bases its foundations on predictive coding [23]. In this model, every neuron (value node) is paired with an error node, where the errors are stored explicitly. Only the value nodes are updated during the learning phase and, during the prediction phase, only the error nodes are updated. [24] relies on a similar mechanism to explicitly store errors. The authors do so by employing dendritic microcircuits and interneurons. The difference in neural activity between the postsynaptic neuron and the interneuron encodes the error. Lastly, [25] introduces the idea of equilibrium propagation, inspired by contrastive learning. By employing local learning rules, and differences of neural activity over time, it can compute gradients similar to those generated by backpropagation.

B. *Biologically plausible neurons model but implausible learning mechanism*

The works in this category propose new methods to train SNN, seeking to achieve the best performance in practice without being limited by biological feasibility. All these approaches consist of a more effective variation of backpropagation suitable for spiking units, which are highly energy efficient. In all

of them, the computations are global. [26] and [27] introduce two similar approaches to learn a temporal spike pattern, which once decoded, represents a specific value. The authors use the number of spikes within an interval as a surrogate for gradient backpropagation. Therefore, they successfully combine rate coding with a temporal credit assignment. The method proposed in [28] also learns a temporal spike pattern. However, in this approach, there is only one spike per interval. Here, the membrane potential at spike times is used as a surrogate for gradient backpropagation. In a similar way, [7] uses the membrane potential to generate the backpropagation of errors. Although in contrast, the used values are picked at fixed times rather than spiking times. The authors in [29] and [30] entirely adopt a temporal coding by choosing the spiking times as a surrogate for gradient backpropagation. In their models, there is only one spike per interval. Recently, [31] proposed an event-based backpropagation, instead of using intervals of time. In this work, the authors combine spike times and membrane voltages as a surrogate for gradient backpropagation.

C. *Biologically plausible learning mechanism and neuron model*

In this last category, we include important works that are entirely biologically feasible. Most of the works in this section build upon the concept *Spike timing dependent plasticity* (STDP) [32], a form of Hebbian learning that strengthens the synaptic connections between two neurons when spike at close time points. The first work in this section is the Tempotron [33], one of the first of its kind. The authors trained a single neuron to recognize specific input spike trains, with local learning rules based on Hebbian learning. The neuron will only spike in the presence of a target pattern. To update the weights in a gradient-based manner, it uses the spike times and the membrane potentials at spike time within a train. Conversely, ReSuMe [34] is an approach based on STDP rules and the Widrow-Hoff rule [35]. It teaches a desired output spiking pattern when an input pattern of spikes is provided to a network. Among more recent works, SuperSpike [36] proposed a gradient descent approach based on the membrane potential of the neurons. The weights changes depend on the postsynaptic neuron's membrane potential and the presynaptic spike train. However, the errors in the networks are still explicitly globally propagated backwards, which should be biologically justified. Nevertheless, the subsequent work addresses this objection. Authors in [37] proposed a similar approach to SuperSpike, but in this case, the network uses a cost function, presented in the work [38], which is local in time and space. This addition overcomes the only (potentially) non-plausible aspect of SuperSpike. Similar to SuperSpike, [39] introduces a gradient descent approach, approximating the backpropagation algorithm with STDP rules. The weights updates only depend on the spikes times of the presynaptic and postsynaptic neurons. It also explicitly globally propagates the errors backwards. Yet, we also included this work in this section for its almost entirely biological feasibility. Besides the

previous works, there are many others dedicated to exploring the effectiveness of the STDP in an unsupervised or rewarded-modulated manner. In this paragraph, we will mention only the most relevant ones. One of the most popular studies dedicated to that is [40]. It applies the STDP rule in a fully unsupervised manner to recognize handwritten digits (MNIST). In their network, they included up to 6400 output neurons, and, after training, only the most active output neuron per class (10 in total) are kept. Inspired by this study, [41] used the same network and similar learning rules, but including a rewarded-modulated modification of STDP. They only employed one output neuron per class and forced the desired output neurons to spike during learning. In this way, the previous approach is modified to enable supervised learning. [42] was again based on the work proposed in [40], this time, aiming for more complex topology and architecture. The authors proposed a multi-layer Spiking Neural Network containing convolutional and pooling layers and trained with STDP rules. This training was done in a layer-by-layer fashion, and in an unsupervised way. However, the network used a support vector machine (SVM) to classify the spikes in the final layer. Yet, it can be considered as a feature extractor. Authors in [43] extended that previous work using rewarded-modulated STDP. As in [41], the authors forced the output neurons to spike. Therefore, they trained the first layers with unsupervised STDP and the last layer with supervised rewarded-modulated STDP. In this way, the network could perform classification with no need for an external classifier. Differently, authors in [44] proposed a learning mechanism based on local Hebbian rules, which relies on structural changes instead of weight updates, and employs binary synapses. Similar to [24], but employing spiking neurons, [45] uses segregated dendrites (multi-compartments neurons) to store errors explicitly in a local manner. In their approach, the difference between feedforward activity and feedback activity is used to compute these local errors. For the weight updates, a local gradient descent approach is used.

III. OUR APPROACH

A. Neuron and synapse model

We use the Leaky Integrate and Fire (LIF) with adaptation as a spiking neuron model. It extends the simple LIF model, one of the most common models in theoretical studies on SNN, by causing the neuron to resist spiking consecutively (for details see Chapter 6.1 of the book “Neural Dynamics” [46]). Computationally, its membrane potential is treated as an internal state. If the synapse is excitatory, the membrane potential increases. Contrary, if the synapse is inhibitory, the membrane potential decreases. When this potential reaches a certain *threshold*, the neuron produces a spike, a sudden increase in its membrane potential followed by a drop to a *reset* value below its threshold. The dynamics of its membrane potential [46] are described by equations (1) and (2).

$$\tau_m \frac{\partial V}{\partial t} = RI(t) - (V - V_{rest}) - Rw_k, \quad (1)$$

$$\tau_k \frac{\partial w_k}{\partial t} = a_k(V - V_{rest}) - w_k + b_k \tau_k \sum \delta(t - t^{(f)}), \quad (2)$$

where τ_m is the membrane time constant, V is the membrane potential, R is the resistance of the membrane, $I(t)$ is the input current at time t , V_{rest} is the resting membrane potential value, w_k is the adaptation variable, τ_k is the adaptation time constant, a is the adaptation voltage coupling variable, b is the adaptation increment variable, $t^{(f)}$ is the firing time of the neuron and δ represents the Dirac delta function, which we employ as indicated in Equation (3).

$$\delta(x) = \begin{cases} 1, & \text{if } x = 0 \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

We used biologically realistic values for all the above parameters [47]–[49]. These values are τ_m : 100.0, τ_k : 40.0, R : 100.0, initial V : -70.0, V_{rest} : -70.0, *reset*: -51.0, w_k : 0.0, *threshold*: -50.0, a : 0.01, b : 0.3. We chose the LIF with adaptation model for our neurons due to its simplicity and effectiveness in modelling realistic neural dynamics [50]. The number of floating-point operations (FLOPS) needed to simulate this model (one neuron) during 1ms is 10 [14]. We can notice how computationally cheap this model is, compared to complex models, such as the Hodgkin-Huxley model [51] that would require 1200 FLOPS to simulate the same scenario. Plus, it does not add excessive complexity compared to the simple LIF model, which would require 5 FLOPS for the same simulation.

In addition, we implemented lateral inhibition when various neurons coexist within a network. With this mechanism, when a neuron spikes, the activity of the neighbouring neurons is reduced. This process can be found in BNNs [52] and generates competitiveness among nearby neurons.

We implement the synapses model as unidirectional non-symmetrical weights, where forward and feedback connections follow the same dynamics. Here, the input current to a postsynaptic neuron is defined in Equation 4.

$$I_j(t) = \sum_i S_i(t)W_{ij}, \quad (4)$$

where $I_j(t)$ is the input current to the postsynaptic neuron j at time t , $S_i(t)$ is the output of the presynaptic neuron i at time t and W_{ij} is the synaptic weight of connecting the presynaptic neuron i and the postsynaptic neuron j . $S_i(t)$ will have the value 1 if the neuron i spikes at time t , and the value 0 otherwise. In addition, the synapses can be excitatory or inhibitory. If the synapse is excitatory, the weight describing it will be positive. If the synapse is inhibitory, the weight describing it will be negative.

B. Coding scheme

In our simulations, the communication between neurons is performed in a clock-driven way. The information exchanged between neurons is encoded in intervals of 200ms of duration, where the time is discretized in 1ms time steps. The encoding of an input value into spikes can be modelled with a Poisson

process [53]. The Poisson distribution used in the encoding process is given in Equation 5.

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}. \quad (5)$$

Regarding the decoding scheme, we introduce a novel coding scheme that combines rate and temporal coding. This method is inspired by the spike counting within an interval (rate coding), adopted by some researchers (e.g. [26], [27]), and the single spike timing within an interval (temporal coding), adopted by other researchers (e.g. [29], [30]). In our approach, we perform a time-weighted spike counting. That means that earlier spikes will contribute more to the decoded result than those later in the interval. Specifically, a spike at the beginning of the interval ($t=0$) contributes twice as much as a spike at the end ($t=199$). Thus, ranging from $\times 1$ to $\times 0.5$, the contribution of every spike depends on its timing within the range. The resulting value is then downscaled by a factor of 100 to match the original value's scale. This decodification is described in Equation 6.

$$Decoded_i = \frac{\sum_{t \in [0, 200)} S_i(t) C_i}{100}, \quad (6)$$

where $Decoded_i$ represents the decoded value of neuron i , S_i represents the spikes train outputted by the neuron i and C_i represents the contributions of the spikes at each time step. $S_i(t)$ takes the value 1 if the neuron i spikes at time t , and the value 0 otherwise. It must be noticed that, due to the randomness introduced by the Poisson-based spike generation, there is a small amount of noise added during the encoding.

C. Errors encoding

Authors in [9] linked together a set of different learning methods that use the difference in neural activities to encode errors. They argued that the brain could also use this technique to compute errors locally. They called this technique Neural Gradient Representation by Activity Differences (NGRAD). In our work, we employ a novel mechanism that may fall into this category. In our network, two neurons are always connected by a forward and a feedback connection. First, the forward connections transmit information to perform predictions. Then, the feedback connections transmit information to adjust weights accordingly and enable learning. As shown in Figure 1, we could compute the error of a neuron as the difference between its activity at two different points on time: (i) after receiving an input current through its forward connections and (ii) after receiving an input current through its feedback connections.

However, we do not use the errors explicitly during the learning process. In networks with non-symmetric feedback weights, these two activities tend to converge to the same value. This phenomenon receives the name of ‘‘feedback alignment’’ [19].

D. Learning rules

As a learning method, a set of Hebbian-based learning rules to achieve local plasticity and computations. Here, the weights updates only depend on the presynaptic and postsynaptic

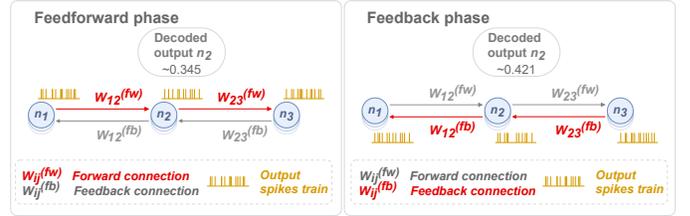


Fig. 1: Example of a decoded output spikes train during the two different phases.

neurons’ activity. Moreover, it does not require any external control or global error during the computations. Having two connected neurons, with output x_1 and x_2 , and the desired target output y in the second neuron, as shown in Figure 2A, we can define the most simplistic rule for Hebbian learning [10] as in Equation 7. If the neurons are spiking units, these values represent their decoded spike trains.

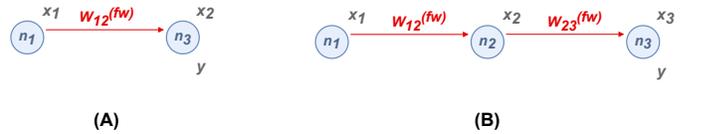


Fig. 2: (A) Two and (B) three neurons connected consecutively by feedforward connections.

$$W_{ij} = x_i y. \quad (7)$$

When progressive updates are desired, we can add a learning rate α and decompose it into an anti-Hebbian rule, defined in Equation 8, and a Hebbian rule, defined in Equation 9.

$$\Delta W_{ij} = -\alpha x_i x_j, \quad (8)$$

$$\Delta W_{ij} = \alpha x_i y. \quad (9)$$

With these rules, we can address the credit assignment problem in single-hidden layer networks. However, they cannot be used in multi-layer neural networks, like the one in Figure 2B, as we do not know the target outputs in the hidden layers.

Target propagation [17], [18] is an alternative to backpropagation that computes a target output in every neuron instead of gradients. Its basic idea is to assign close values in the presynaptic neurons that hopefully lead to the desired values in the postsynaptic neurons. In our work, we mimic this idea with a much more simplistic approach. We introduce a novel method to approximate the target propagation’s main idea making use of the feedback alignment theory. For that, we first include feedback connections in our network. Then, in order to compute the target outputs in the hidden layers, the postsynaptic neurons forward their desired outputs (label) back through their feedback connections. Then, according to the feedback alignment theory [19], the neurons’ output when performing feedforward will tend to converge to the neurons’ output when performing feedback. Therefore, in a situation like the one shown in Figure 3, the value x_2 will tend to come

closer to y_2 . Notice that x_1 , x_2 and x_3 refer to the decoded output spike trains during the feedforward phase, y_3 refers to the desired decoded output value (label) and y_2 refers to the decoded output spike trains during the feedback phase. In this case, y_2 is obtained by propagating y_3 through the feedback connections.

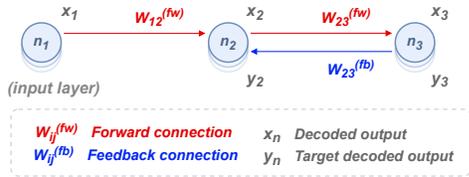


Fig. 3: Three layers of neurons connected consecutively by feedforward connections and feedback connections, where the first layer is the input layer with no feedback connections.

In the same way, the feedback weights also need to be modified. When, for instance, a higher input current is desired in the postsynaptic neuron, the influence of excitatory presynaptic neurons (positive synaptic weights) should be increased, and the influence of inhibitory presynaptic neurons (negative synaptic weights) should be reduced. According to the feedback alignment theory, we can indirectly do this by increasing the feedback weights with the excitatory neurons, and reducing the feedback weights with the inhibitory neurons. We implement these feedback weights updates with a local anti-Hebbian and a local Hebbian based rules to accomplish biological plausibility. These rules are described by equations 10 and 11, using the same notation as in Figure 3.

$$\Delta W_{ij}^{(fb)} = \frac{-\alpha x_i y_j}{W_{ij}^{(fw)}}, \quad (10)$$

$$\Delta W_{ij}^{(fb)} = \frac{\alpha y_i y_j}{W_{ij}^{(fw)}}. \quad (11)$$

Equation 10 describes an update while the information is forwarded back. Equation 11 describes an update once the information is sent back. For implementation purposes, they can be abstracted into one single rule, given in Equation 12.

$$\Delta W_{ij}^{(fb)} = \frac{\alpha(y_i - x_i)y_j}{W_{ij}^{(fw)}}. \quad (12)$$

Notice that $W_{ij}^{(fw)}$, which represent the synaptic weight of the forward connection, is included to take into account the type of synapse in the update (excitatory or inhibitory) through its sign. It works as a scaling factor too. $W_{ij}^{(fw)}$ is a local value to both the presynaptic neuron i and the postsynaptic neuron j , so, in biology, it could influence updates in $W_{ij}^{(fb)}$. Also, combining Equations 8 and 9 and adjusting its notation, we obtain Equation 13.

$$\Delta W_{ij}^{(fw)} = \alpha x_i (y_j - x_j). \quad (13)$$

Hence, our learning mechanism consists of two phases, i.e. feedforward phase and feedback phase. During the feedforward phase, the feedback weights do not transmit information,

and the forward weights are updated using the learning rule described in Equation 13. During the feedback phase, the forward weights do not transmit information, and the feedback weights are updated using the learning rule described in Equation 12. Using this set of rules, there is no need for a loss function and its optimization. These alternative provides more biological realism than, for instance, backpropagation, which is based on mathematical optimization concepts.

E. Adaptive learning rate factor

Here, we also introduce a novel adaptive learning rate technique that drastically speeds up our simulations. It uses a learning rate value per synaptic weight, and variates according to the alignment between the forward neural activity and the feedback neural activity, i.e. x_n and y_n in Figure 3. After every feedback phase, if x_n and y_n have closer values than in the previous feedback phase, then the learning rate of the weight that connects to the neuron n is increased. Conversely, if x_n and y_n have more distant values, the learning rate decreases. This technique could be compared with the momentum algorithm [54] for ANNs. We empirically found that a 2% increase/decrease in the individual learning rates when the above conditions are met is the optimal value. Since after some point during learning (when more detailed updates are needed), x_n and y_n stop coming closer, the learning rate factors start decreasing considerably. Therefore, these learning rate factors will tend to adopt small values at the end of the training.

IV. EXPERIMENTAL PROCEDURE AND DISCUSSION

We conducted several experiments to evaluate our approach. In our networks, the first layer is in charge of encoding the input values into trains of spikes. Therefore, there are no feedback connections between the first and the second layer. Also, we did not explore complex topologies and only fully connected layers are used here. The forward and feedback weights are initialized randomly using a Gaussian distribution with a mean of 2.5 and a standard deviation of 1.0.

A. Boolean problems

While the AND and OR problems are easy to solve, the XOR problem poses some difficulty due to its non-linearity. This non-linearity makes it impossible to solve employing a single layer perceptron and, hence, a multi-layer extension is needed [55]. The boolean operations are shown in Table I. In our simulation, we used the value 0.01 instead of 0 to activate our LIF neurons.

TABLE I: Boolean problems

		AND	OR	XOR
a	b	$a \cdot b$	$a + b$	$a \oplus b$
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

The architecture employed in these problems contains four layers with 2-128-128-2 neurons. The starting learning rate

was 0.001, and the amount of lateral inhibition was 5mV. We let the models train for 20 epochs, though they could successfully learn the tasks in just a few epochs, as shown in Figure 4. With these results we proved that a multi-layer SNN with feedback connections trained with our learning method can solve simple linear and non-linear tasks perfectly.

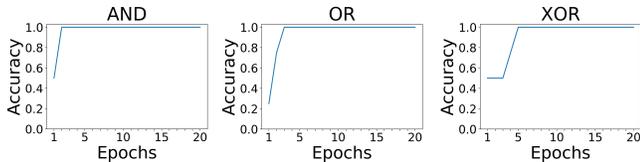


Fig. 4: Training history of the tasks AND, OR and XOR.

B. Autoencoding task

In this task, we trained an autoencoder to learn a compressed representation of an image. For that, we used the MNIST dataset with normalized values. We did not use validation set to early stop the model. Instead, we used the training set to train our network until the training performance stops improving (5 epochs), and then test over the test set. The employed architecture contains 784-64-784 neurons, with a starting learning rate of 0.001 and a lateral inhibition of 1mV. We used Mean Squared Error (MSE) to measure the performance of the model. The resulting MSE over both the training and test set is 0.037. Notice that these values are computed with the normalized data. In Figure 5, we display some images from the test set after encoding and decoding it using our trained network. We can notice that the reconstructions closely resemble the original images. In addition, we trained a regular ANN with the same architecture using backpropagation for comparison purposes during 100 epochs. It reached an MSE of 0.0138 in both the training and test set. We can notice that the ANN performs better than our SNN. Something not surprising. Even the state-of-the-art approaches to biologically plausible learning are still far from performing comparably to ANNs trained with backpropagation. In the next task, we perform a comparison between state-of-the-art approaches to biologically plausible learning and an ANN trained with backpropagation, among others. Despite the difference, the performance measurement (MSE) of our approach and the ANN trained with backpropagation are in the same order of magnitude ($0.0X$) in this task, while the model was trained with considerably less epochs.



Fig. 5: Examples of reconstructed images from the test set with our trained neural network. Top row: original images. Bottom row: reconstructed images (encoded and decoded).

C. Handwritten digits recognition

In this task, we trained various multi-layer neural networks to perform handwritten digits recognition. Again, we used the MNIST dataset with normalized values, and we did not use validation set to early stop the model. Instead, we used the training set to train our network until the training performance stops improving (5 epochs), and then test over the test set. The starting learning rate of 0.001 and a lateral inhibition of 1mV. A comparison between different configurations and their performance can be found in Table II.

TABLE II: Comparison of different architectures over the MNIST dataset. Each model was trained for 5 epochs.

Neurons per layer	Training accuracy	Test accuracy
784-200-10	90.0	90.3
784-400-10	90.9	91.1
784-1200-10	<u>91.8</u>	<u>91.7</u>
784-3000-10	91.5	91.2
784-200-200-10	89.9	90.5
784-1500-1000-500-10	88.9	89.0

We also compare our approach with others entirely biologically plausible learning methods in Table III. While some architectures provide small improvements compared to others, there seems to be a limit to learning. We hypothesise that it is due to the stochastic encoding process. We elaborate on this idea in the next section. We also consider in the comparison a multi-layer ANN trained using backpropagation. We did not tune the hyperparameters of this last network. Instead, we adopted the same parameters as with the best performing model in our approach: three layers with 784-1200-10 neurons and starting learning rate of 0.001 (Adam optimizer). The activation function in the last layer is a softmax, and ReLU in the hidden layers. The loss function used is MSE.

V. CONCLUSION AND FUTURE WORK

Current models of ANNs differ from BNNs in many aspects. Researchers are working to bridge this gap. However, there are still plenty of unknown details about the brain and the computation it performs. Seeking this goal, we present an innovative approach, constrained by biological limitations. This work intends to give a step towards brain-inspired neural networks rather than being a conclusive work. Our proposed learning method bases its foundations on recent brain-inspired elements: local plasticity rules, target propagation, feedback alignment and realistic models of neurons and synapses. In the same way, our learning mechanism does not explicitly use errors and does not minimize a loss function. Consequently, it achieves more biological realism than other mathematical optimization-based solutions. We also introduce an adaptive learning rate technique that drastically reduces the training time. In all the tested tasks, the maximum performance is

TABLE III: Comparison with the state-of-the-art biologically plausible approaches. A “-” indicates that the value is not provided in the paper.

Work	Description	Test acc. MNIST
ANN (this paper)	3-layers fully connected ANN trained using backpropagation. For comparison purposes.	98.5
<i>STDP based approaches</i>		
F. Ponulak et al. [34]	<i>ReSuMe</i> learning algorithm. Supervised learning. It employs STDP and anti-STDP rules.	-
A. Tavanaei et al. [39]	Supervised learning. It employs a locally approximated backpropagation using STDP and anti-STDP rules.	97.2
P. U. Diehl et al. [40]	Unsupervised learning. It employs a modified STDP rule and many neurons in the output layer. Then those output neurons are filtered to only 10.	95.0
Y. Hao et al. [41]	Supervised learning. It employs a rewarded-modulated STDP rule.	96.7
M. Mozafari et al. [43]	Supervised learning. It employs a rewarded-modulated STDP rule and a deep convolutional architecture	97.2
<i>Non-STDP based approaches</i>		
S. Hussain et al. [44]	Supervised learning. It is based on Hebbian learning with binary synapses. Instead of update the weights, it performs structural modifications in the network.	90.3
R. Gütig et al. and Zhao et al. [33], [56]	<i>Tempotron</i> learning algorithm. Supervised learning. It is based on Hebbian learning, and it is designed to classify spatiotemporal patterns of spikes.	91.3
J. Guerguiev et al. [45]	Supervised learning. It uses dendritic compartments in the neurons to compute local errors. It employs a local version of gradient descent.	96.3
F. Zenke et al. [36]	<i>SuperSpike</i> learning algorithm. Supervised learning. It employs local Hebbian learning rules. It uses a gradient descent approach based on the neurons’ membrane potential.	-
Ours (this paper)	Supervised learning. It employs Hebbian learning. It is based on target backpropagation and feedback alignment to enable learning. Errors can be computed locally as a difference between neural activities.	91.7

reached within the first 5 epochs, unlike some state-of-the-art approaches (e.g. the approach in [39] employs around 1200 epochs). Our approach was able to successfully solve a set of linear and non-linear tasks and an image autoencoding task. Furthermore, our approach performs comparably to other biologically plausible non-STDP-based works in the handwritten digits recognition task. On the other hand, the used coding scheme, time-weighted spike counting, is not the optimal one. Since the encoding is based on a Poisson random process, the resulting scheme is stochastic. It introduces randomness during the learning process, which leads to the use of non-accurate information during the weights updates. While other learning mechanisms considered in the comparison achieves high performance over the training set and moderate performance over the test set (e.g. Zhao et al. [56] achieve 99.36% acc. over the training set and 91.3% acc over the test set), our approach seems to have a limit in the learning. During training, very detailed and accurate weight updates are needed after a certain point. This added noise keeps the network from achieving the needed detailed updates. We hypothesize that this is the reason why, as shown in Table II, the networks seem to reach a limit in learning around the same performance measurement (for both training and test). A biologically plausible alternative to this coding scheme is to use a temporal encoding with only one spike per spikes train. That is, the timing of individual spikes. With this alternative, information can be accurately encoded and decoded, with the values represented by one spike through the temporal dimension. We are confident that replacing the current stochastic encoding/decoding scheme with a highly accurate temporal scheme would lead to very improved efficiency. In addition, evaluating the proposed learning mechanism with more complex topologies (convolutional, recurrent, etc) or architectures (deeper networks, ResNet, UNet, etc) and over more complex tasks (object recognition, medical image segmentation, etc) would be desirable as well.

REFERENCES

- [1] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
- [2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [3] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.
- [4] D. G. Barrett, A. S. Morcos, and J. H. Macke, “Analyzing biological and artificial neural networks: challenges with opportunities for synergy?” *Current opinion in neurobiology*, vol. 55, pp. 55–64, 2019.
- [5] R. Geirhos, C. R. M. Temme, J. Rauber, H. H. Schütt, M. Bethge, and F. A. Wichmann, “Generalisation in humans and deep neural networks,” *arXiv preprint arXiv:1808.08750*, 2018.
- [6] D. Pospisil, A. Pasupathy, and W. Bair, “Comparing the brain’s representation of shape to that of a deep convolutional neural network,” in *Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS)*, 2016, pp. 516–523.
- [7] J. H. Lee, T. Delbruck, and M. Pfeiffer, “Training deep spiking neural networks using backpropagation,” *Frontiers in neuroscience*, vol. 10, p. 508, 2016.

- [8] Y. Wu, L. Deng, G. Li, J. Zhu, and L. Shi, "Spatio-temporal backpropagation for training high-performance spiking neural networks," *Frontiers in neuroscience*, vol. 12, p. 331, 2018.
- [9] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton, "Backpropagation and the brain," *Nature Reviews Neuroscience*, vol. 21, no. 6, pp. 335–346, 2020.
- [10] D. O. Hebb, *The organization of behavior: A neuropsychological theory*. Psychology Press, 2005.
- [11] C. F. Cadieu, H. Hong, D. L. Yamins, N. Pinto, D. Ardila, E. A. Solomon, N. J. Majaj, and J. J. DiCarlo, "Deep neural networks rival the representation of primate it cortex for core visual object recognition," *PLoS Comput Biol*, vol. 10, no. 12, p. e1003963, 2014.
- [12] J. C. Whittington and R. Bogacz, "Theories of error back-propagation in the brain," *Trends in cognitive sciences*, vol. 23, no. 3, pp. 235–250, 2019.
- [13] Y. Bengio, D.-H. Lee, J. Bornschein, T. Mesnard, and Z. Lin, "Towards biologically plausible deep learning," *arXiv preprint arXiv:1502.04156*, 2015.
- [14] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE transactions on neural networks*, vol. 15, no. 5, pp. 1063–1070, 2004.
- [15] P. R. Roelfsema and A. Holtmaat, "Control of synaptic plasticity in deep cortical networks," *Nature Reviews Neuroscience*, vol. 19, no. 3, p. 166, 2018.
- [16] T. Gollisch and M. Meister, "Rapid neural coding in the retina with relative spike latencies," *science*, vol. 319, no. 5866, pp. 1108–1111, 2008.
- [17] Y. Le Cun, "Learning process in an asymmetric threshold network," in *Disordered systems and biological organization*. Springer, 1986, pp. 233–240.
- [18] D.-H. Lee, S. Zhang, A. Fischer, and Y. Bengio, "Difference target propagation," in *Joint european conference on machine learning and knowledge discovery in databases*. Springer, 2015, pp. 498–515.
- [19] T. P. Lillicrap, D. Cownden, D. B. Tweed, and C. J. Akerman, "Random synaptic feedback weights support error backpropagation for deep learning," *Nature communications*, vol. 7, no. 1, pp. 1–10, 2016.
- [20] M. A. Carreira-Perpinan and G. E. Hinton, "On contrastive divergence learning," in *Aistats*, vol. 10. Citeseer, 2005, pp. 33–40.
- [21] R. C. O'Reilly, "Biologically plausible error-driven learning using local activation differences: The generalized recirculation algorithm," *Neural computation*, vol. 8, no. 5, pp. 895–938, 1996.
- [22] J. C. Whittington and R. Bogacz, "An approximation of the error backpropagation algorithm in a predictive coding network with local hebbian synaptic plasticity," *Neural computation*, vol. 29, no. 5, pp. 1229–1262, 2017.
- [23] R. P. Rao and D. H. Ballard, "Predictive coding in the visual cortex: a functional interpretation of some extra-classical receptive-field effects," *Nature neuroscience*, vol. 2, no. 1, pp. 79–87, 1999.
- [24] J. Sacramento, R. P. Costa, Y. Bengio, and W. Senn, "Dendritic cortical microcircuits approximate the backpropagation algorithm," *arXiv preprint arXiv:1810.11393*, 2018.
- [25] B. Scellier and Y. Bengio, "Equilibrium propagation: Bridging the gap between energy-based models and backpropagation," *Frontiers in computational neuroscience*, vol. 11, p. 24, 2017.
- [26] S. B. Shrestha and G. Orchard, "Slayer: Spike layer error reassignment in time," *arXiv preprint arXiv:1810.08646*, 2018.
- [27] J. Wu, Y. Chua, M. Zhang, Q. Yang, G. Li, and H. Li, "Deep spiking neural network with spike count based learning rule," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–6.
- [28] S. M. Bohte, J. N. Kok, and H. La Poutre, "Error-backpropagation in temporally encoded networks of spiking neurons," *Neurocomputing*, vol. 48, no. 1–4, pp. 17–37, 2002.
- [29] S. R. Kheradpisheh and T. Masquelier, "Temporal backpropagation for spiking neural networks with one spike per neuron," *International Journal of Neural Systems*, vol. 30, no. 06, p. 2050027, 2020.
- [30] I. M. Comsa, K. Potempa, L. Versari, T. Fischbacher, A. Gesmundo, and J. Alakuijala, "Temporal coding in spiking neural networks with alpha synaptic function," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 8529–8533.
- [31] T. C. Wunderlich and C. Pehle, "Event-based backpropagation can compute exact gradients for spiking neural networks," *Scientific Reports*, vol. 11, no. 1, pp. 1–17, 2021.
- [32] J. Sjöström and W. Gerstner, "Spike-timing dependent plasticity," *Scholarpedia*, vol. 5, no. 2, p. 1362, 2010, revision #184913.
- [33] R. Gütiğ and H. Sompolinsky, "The tempotron: a neuron that learns spike timing-based decisions," *Nature neuroscience*, vol. 9, no. 3, pp. 420–428, 2006.
- [34] F. Ponulak and A. Kasiński, "Supervised learning in spiking neural networks with resume: sequence learning, classification, and spike shifting," *Neural computation*, vol. 22, no. 2, pp. 467–510, 2010.
- [35] B. Widrow and M. E. Hoff, "Adaptive switching circuits," Stanford Univ Ca Stanford Electronics Labs, Tech. Rep., 1960.
- [36] F. Zenke and S. Ganguli, "Superspike: Supervised learning in multilayer spiking neural networks," *Neural computation*, vol. 30, no. 6, pp. 1514–1541, 2018.
- [37] J. Kaiser, H. Mostafa, and E. Neftci, "Synaptic plasticity dynamics for deep continuous local learning (decolle)," *Frontiers in Neuroscience*, vol. 14, p. 424, 2020.
- [38] H. Mostafa, V. Ramesh, and G. Cauwenberghs, "Deep supervised learning using local errors," *Frontiers in neuroscience*, vol. 12, p. 608, 2018.
- [39] A. Tavanaei and A. Maida, "Bp-stdp: Approximating backpropagation using spike timing dependent plasticity," *Neurocomputing*, vol. 330, pp. 39–47, 2019.
- [40] P. U. Diehl and M. Cook, "Unsupervised learning of digit recognition using spike-timing-dependent plasticity," *Frontiers in computational neuroscience*, vol. 9, p. 99, 2015.
- [41] Y. Hao, X. Huang, M. Dong, and B. Xu, "A biologically plausible supervised learning method for spiking neural networks using the symmetric stdp rule," *Neural Networks*, vol. 121, pp. 387–395, 2020.
- [42] S. R. Kheradpisheh, M. Ganjtabesh, S. J. Thorpe, and T. Masquelier, "Stdp-based spiking deep convolutional neural networks for object recognition," *Neural Networks*, vol. 99, pp. 56–67, 2018.
- [43] M. Mozafari, M. Ganjtabesh, A. Nowzari-Dalini, S. J. Thorpe, and T. Masquelier, "Combining stdp and reward-modulated stdp in deep convolutional spiking neural networks for digit recognition," *arXiv preprint arXiv:1804.00227*, vol. 1, 2018.
- [44] J. Hussain, S.-C. Liu, and A. Basu, "Improved margin multi-class classification using dendritic neurons with morphological learning," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2014, pp. 2640–2643.
- [45] J. Guerguiev, T. P. Lillicrap, and B. A. Richards, "Towards deep learning with segregated dendrites," *ELife*, vol. 6, p. e22901, 2017.
- [46] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.
- [47] R. Naud, N. Marcille, C. Clopath, and W. Gerstner, "Firing patterns in the adaptive exponential integrate-and-fire model," *Biological cybernetics*, vol. 99, no. 4, pp. 335–347, 2008.
- [48] L. Badel, S. Lefort, R. Brette, C. C. Petersen, W. Gerstner, and M. J. Richardson, "Dynamic iv curves are reliable predictors of naturalistic pyramidal-neuron voltage traces," *Journal of Neurophysiology*, vol. 99, no. 2, pp. 656–666, 2008.
- [49] C. Clopath, R. Jolivet, A. Rauch, H.-R. Lüscher, and W. Gerstner, "Predicting neuronal activity with simple models of the threshold type: Adaptive exponential integrate-and-fire model with two compartments," *Neurocomputing*, vol. 70, no. 10–12, pp. 1668–1673, 2007.
- [50] R. Jolivet, A. Rauch, H.-R. Lüscher, and W. Gerstner, "Integrate-and-fire models with adaptation are good enough," Tech. Rep., 2006.
- [51] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of physiology*, vol. 117, no. 4, pp. 500–544, 1952.
- [52] B. Bridgeman, "Metacontrast and lateral inhibition," *Psychological review*, vol. 78, no. 6, p. 528, 1971.
- [53] M. N. Shadlen and W. T. Newsome, "The variable discharge of cortical neurons: implications for connectivity, computation, and information coding," *Journal of neuroscience*, vol. 18, no. 10, pp. 3870–3896, 1998.
- [54] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural networks*, vol. 12, no. 1, pp. 145–151, 1999.
- [55] L. Noriega, "Multilayer perceptron tutorial," *School of Computing, Staffordshire University*, 2005.
- [56] B. Zhao, R. Ding, S. Chen, B. Linares-Barranco, and H. Tang, "Feed-forward categorization on aer motion events using cortex-like features in a spiking neural network," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 9, pp. 1963–1978, 2014.