

Computational Neuroscience

Project No. 3 Report

Name: Mahroo Hajimehdi

Student No: 610399198

Introduction

The report sets out to achieve three pivotal objectives within the realms of information processing and machine learning, utilizing the Brian2 library for the implementation of simulations and focusing on a dataset comprised of grayscale images. The initial objective centers on exploring the various methods of information encoding and the transformation of input stimuli to spikes. This exploration is not only aimed at understanding the diverse strategies for efficiently encoding information but also at examining the processes that convert real-world inputs, specifically grayscale images, into digital signals or spikes. The use of Brian2 simulations will facilitate an in-depth analysis of how these encoding techniques and spike transformations are applied to grayscale image data.

Moving on to the second objective, the focus shifts towards unsupervised learning and reinforcement learning frameworks. The primary aim here is to gain insights into unsupervised learning algorithms that discover hidden patterns and structures within unlabeled grayscale image data. Additionally, the report will delve into the principles of reinforcement learning, emphasizing how systems can learn and refine their decision-making processes through interactions with an environment that provides feedback. Brian2 simulations will be instrumental in modeling these learning paradigms, particularly showcasing how they can be applied to analyze and learn from datasets comprised of grayscale images.

The final objective addresses the learning processes governed by Hebbian theory, highlighting its fundamental role in neuroscience and neural network training. This section will elaborate on Hebbian theory's basic principles, especially as they pertain to the strengthening of neural connections based on activity levels, and their significance in learning and memory formation with a focus on grayscale images. By employing Brian2 for simulations, this part of the report will provide hands-on analysis and demonstration of Hebbian learning mechanisms as they apply to the dataset of grayscale images.

Throughout the objectives, the Brian2 library stands out as an essential tool for conducting simulations that blend theoretical insights with practical application, specifically tailored to the dataset of grayscale images. This approach aims to deepen the understanding of encoding strategies, learning algorithms, and neural network training processes, facilitating a comprehensive exploration of these areas through the lens of grayscale image data (**Figure 1**).

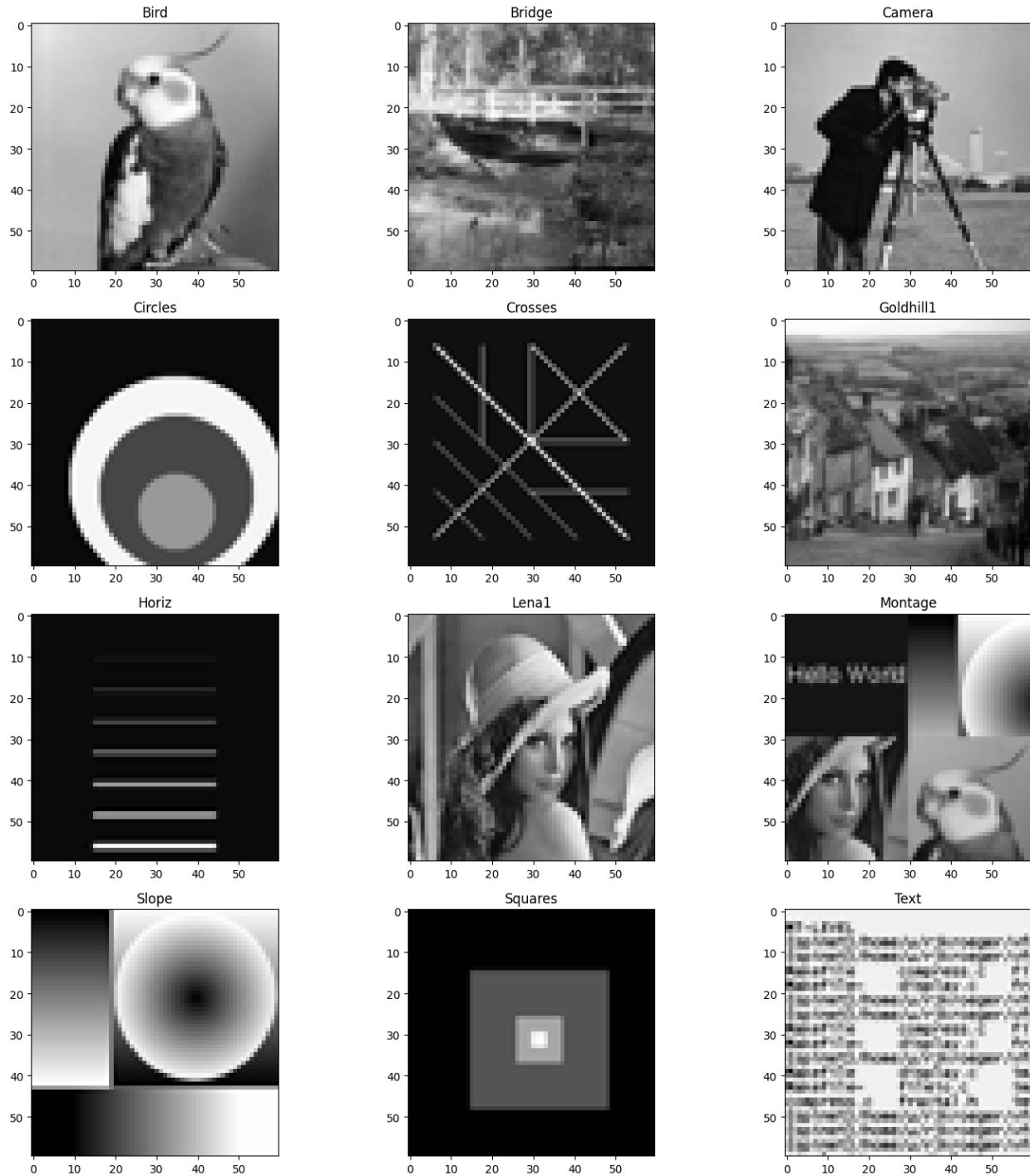


Figure 1 Encoders grayscale image dataset.

Task 1: Implementation of Encoding Methods

The following encoding methods have been implemented and applied to input stimuli present for a duration of T milliseconds.

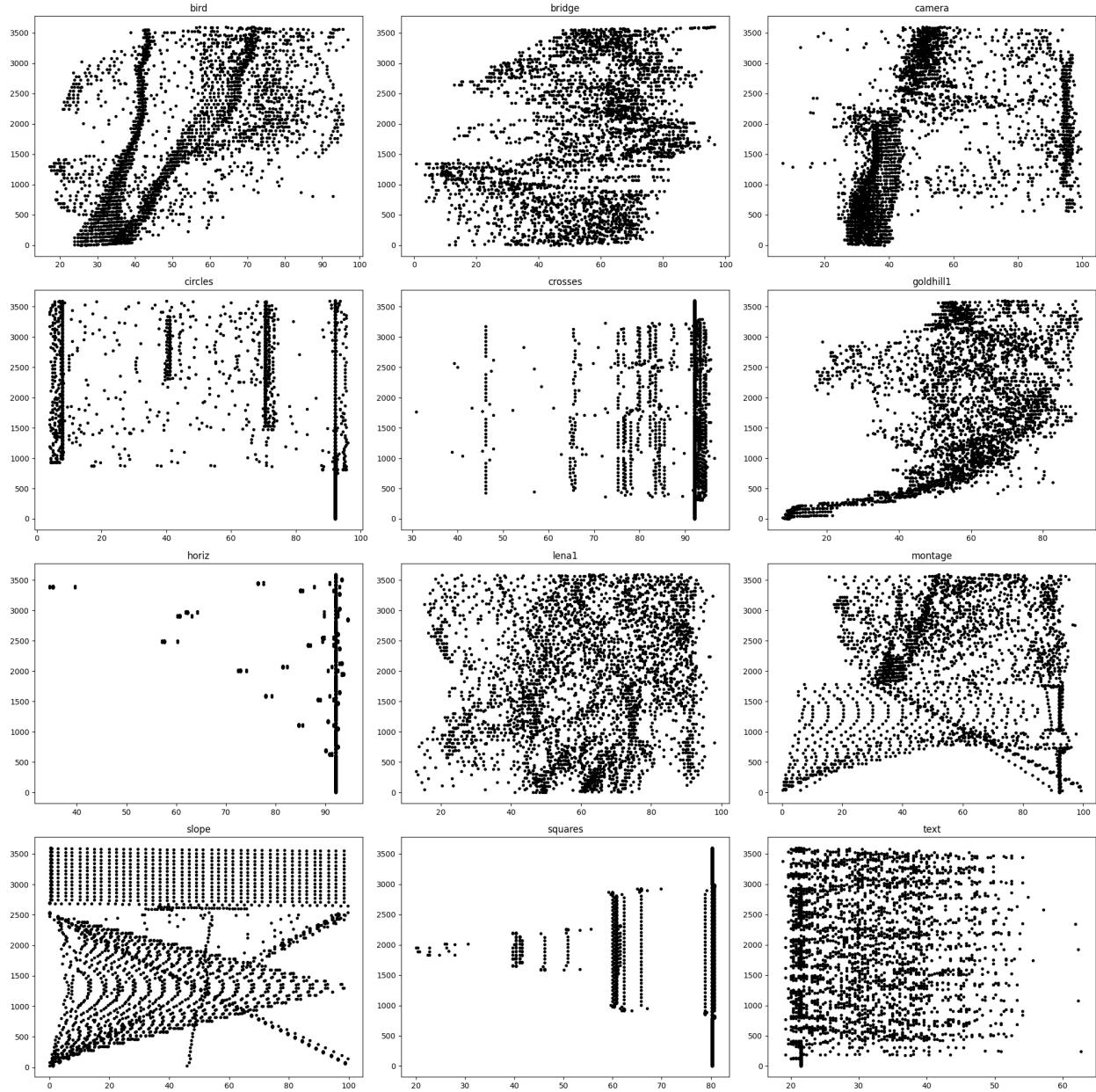


Figure 2 TTFS encoding results. Each neuron represents a pixel in image with its spike delay is proportional to pixel intensity.

A. Time-to-First-Spike Encoding Method

B. Encoding of Numerical Values (Rate Encoding)

C. Poisson Distribution-Based Encoding

The input images have been downsampled from a resolution of 256x256 to 60x60. Each pixel within these images is represented as a neuron. The dataset can be seen in **Figure 1**.

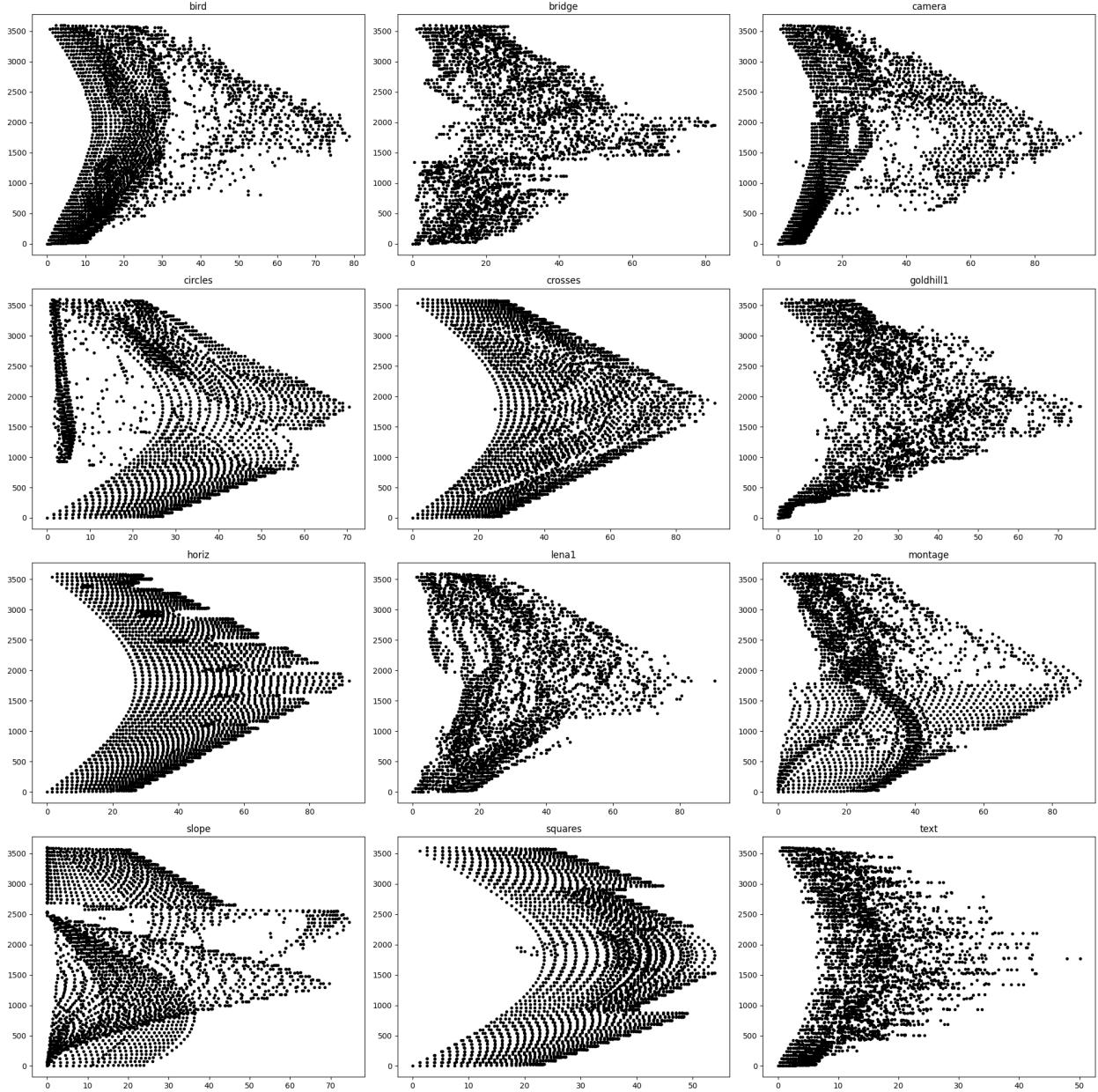


Figure 3 Position encoding. Spike times are inversely related to the normalized distance.

A. Time-to-First-Spike Encoding Method

The Time-To-First-Spike (TTFS) method is a technique used in spiking neural networks (SNNs) where the information is encoded in the timing of the first spike relative to the onset of a stimulus. This method leverages the temporal dynamics of neural responses, encoding information more efficiently by using the time until the first spike to represent signal intensity or other features.

In the implementation, the TTFS method is applied by first normalizing image values to a range between 0 and 1. This normalization step is crucial because it ensures that all pixel intensities are

comparatively scaled, enabling a consistent mapping from intensity to spike timing. Higher intensities are inverted to correspond to earlier spike times, embodying the principle that more intense stimuli trigger quicker neural responses. To apply this in a computational model, intensities are converted to spike times—where a 0 intensity means no spike is generated, adhering to a threshold-based firing mechanism typical in SNNs.

The spike times are then flattened, converting the 2D representation of an image into a 1D list of spike times for each neuron. This step is essential for simplifying the input for spiking neural networks, allowing each neuron in the network to correspond to a particular pixel's intensity (or its derived spike time).

Finally, these spike times are utilized to generate spikes within a SNN using the SpikeGeneratorGroup module in the Brian2 library. This module facilitates the creation of neurons that fire based on predefined spike times, integrating the TTFS-encoded image data into the network's dynamics. The visualization of this encoding process, as seen in **Figure 2**, provides insight into how each image's information is represented within the network, showcasing the efficacy and the efficiency of the TTFS method in encoding and processing visual stimuli in spiking neural networks.

B. Encoding of Numerical Values (Position Encoding)

This method encodes the positional information of pixels within images for use in spiking neural networks (SNNs), using only their relative distances from the image center to determine their spike times. It disregards the intensity information of the pixels, focusing purely on their spatial distribution. As for the implementation it calculates each pixel's distance from the center of this standard image size. This is done using a mesh grid that represents the pixel positions and applying the Euclidean distance formula to find each pixel's distance from the center. Then the distances are normalized by dividing by the maximum found distance, ensuring that all values are between 0 and 1.

Spike times are inversely related to the normalized distance: pixels closer to the center (smaller distance) have shorter spike times, and those further away have longer spike times. This is modeled by calculating the spike time as a function of the normalized distance, with a predefined maximum spike time. The precise moments each "neuron" (pixel) fires are determined by the calculated spike times, effectively encoding positional information into a temporal pattern of spikes (**Figure 3**).

C. Poisson Distribution-Based Encoding

Poisson distribution coding in the context of spiking neural networks (SNNs) is a methodology used for encoding input data based on the principles of stochastic processes, specifically the Poisson process. It represents input signals through the firing rates of neurons, where the rate of firing is randomly determined, following a Poisson distribution. This method is particularly effective for mimicking the randomness observed in biological neural systems, making it a suitable approach for tasks involving sensory data processing or pattern recognition.

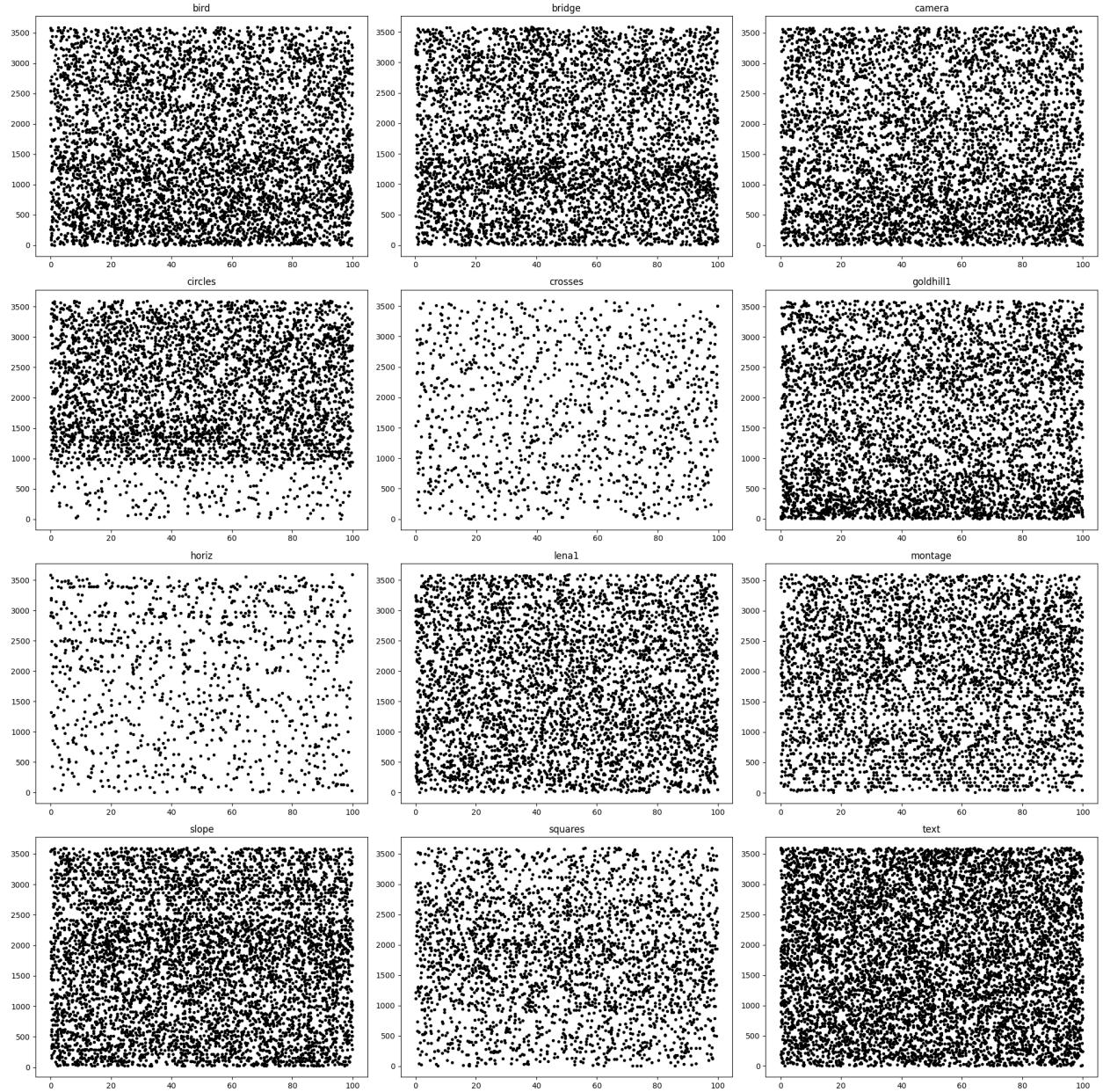


Figure 4 Poisson distribution encoding. Neurons firing rates are proportional to its corresponding pixel intensity.

In implementing Poisson distribution coding with an emphasis on visual stimuli, the process begins by quantifying the maximum firing rate. This rate is directly correlated with the brightest pixel within the image, symbolizing the highest intensity level that a neuron can represent. The firing rates for all neurons are then determined based on the relative intensity of each pixel in the image. Essentially, each neuron is assigned a firing rate that proportionally reflects the intensity of its corresponding pixel, with brighter pixels inducing higher rates (**Figure 4**).

The practical steps of implementation in a computational model, specifically using the Brian2 library, involve the following:

1. Normalization and Rate Calculation: After identifying the maximum intensity within the image, the intensities of all pixels are normalized against this value. The firing rate for each neuron is calculated by mapping the normalized pixel intensity to a corresponding rate, ensuring that neurons representing brighter pixels fire more frequently.
2. Defining the Neuron Group: With the firing rates determined, a neuron group is defined using the 'PoissonGroup' function in Brian2. This function creates a group of neurons that fire according to Poisson statistics, with the firing rates previously calculated. The 'PoissonGroup' function takes the number of neurons (matching the number of pixels or the adjusted number post-image processing) and their rates as inputs, setting up a spiking neural network layer where each neuron's activity statistically reflects the visual input's intensity pattern.
3. Simulation Execution: The simulation is run, allowing the Poisson-coded neural group to generate spikes. Throughout this phase, the stochastic nature of the Poisson process ensures that the firing patterns of the neurons are random but with average rates corresponding to the input image's intensity distribution.

Analysis

Comparing the three encoding schemes, we can note the following key differences:

1. TTFS encoding maps pixel intensities to temporal spike times, while positional coding uses spatial positions, and Poisson coding represents intensities through point densities.
2. TTFS and positional coding preserve explicit structural information about the encoded images, whereas Poisson coding represents intensities through statistical point distributions.
3. TTFS encoding is inherently temporal, capturing image information over time, while positional coding and Poisson coding are spatial representations.
4. The visual patterns and distributions of points/spikes differ significantly among the three encoding schemes, reflecting their unique approaches to representing the same input images.

In *TTFS encoding*, each pixel's intensity value is mapped to the time it takes for a simulated neuron to fire its first spike from an initial resting state. Brighter pixels result in earlier spike times, while darker pixels lead to later or no spikes.

The x-axis represents the discretized time steps, and each vertical line or dot at a specific time point corresponds to a neuron firing its first spike at that moment, encoding the intensity of a particular pixel location.

By observing the distribution of spikes across time, we can discern the encoded shapes and intensity variations within each image. Dense clusters of early spikes represent brighter regions, whereas sparse later spikes encode darker areas.

When comparing the original image to the Time-To-First-Spike (TTFS) pattern, it becomes evident that darker images such as 'Horiz', 'Square', and 'Crosses' generate fewer spikes overall. In contrast, the image labeled 'Lena1' produces a higher number of spikes. The 'Montage' image is particularly noteworthy: upon closer examination, we can discern the spike patterns corresponding to each of the images that it comprises. Notably, the spike pattern of 'Slope' is visible in the bottom right quarter of the 'Montage' image.

The *Poisson distribution* coding technique maps the intensity of each pixel to the density or concentration of points at that spatial location. Brighter pixels are represented by a higher density or clustering of points, while darker pixels have a sparser distribution of points.

When comparing the original image to the pattern resulting from the Poisson distribution, it is observed that darker images labeled as 'Horiz', 'Square', and 'Crosses' exhibit a sparser distribution of points overall. Specifically, the top portion of the 'Circle' image is darker, leading to a sparser distribution of points observed among the initial set of neurons. Conversely, the top portion of 'Goldhill1' is brighter, which accounts for the denser distribution observed in the spiking activity of the first neurons.

In *Positional coding* of a grayscale image, as previously mentioned, spike times are inversely proportional to the normalized distance from the center of the image: pixels closer to the center, which have a smaller normalized distance, generate spikes earlier, while those further from the center produce spikes later. Consequently, the neurons representing the edges of the image emit spikes sooner than those representing the center. This phenomenon accounts for the triangular shape observed in the spike distribution pattern.

Also, the density of spikes is directly related to the intensity of pixels in the image:

1. Brighter Pixels Lead to Higher Spike Densities: In a grayscale image, brighter pixels represent higher intensity values ('Text', 'Lena1'). When these are encoded using positional coding, these higher intensity areas would correspond to a higher density of spikes. This is because the brightness is interpreted as a higher stimulus or a more significant feature, which in the context of many neural coding schemes, translates to increased firing rates or more densely packed spikes.
2. Darker Pixels Result in Lower Spike Densities: Conversely, darker pixels, indicative of lower intensity values, would result in a sparser distribution of spikes. These areas are considered to be of lower stimulus strength, leading to fewer spikes being generated to represent these portions of the image.
3. Grayscale Gradient Representation: For areas of the image with a gradient (i.e., a smooth transition from dark to light), you would observe a correspondingly gradual change in the spike density ('Squares'). This gradient represents a nuanced encoding of the image, allowing for the detailed representation of shapes, edges, and textures within the neural coding framework..

These encoding techniques demonstrate different strategies for translating grayscale image data into spiking neural representations, each with its own strengths and characteristics in terms of preserving spatial, temporal, or statistical information about the original image.

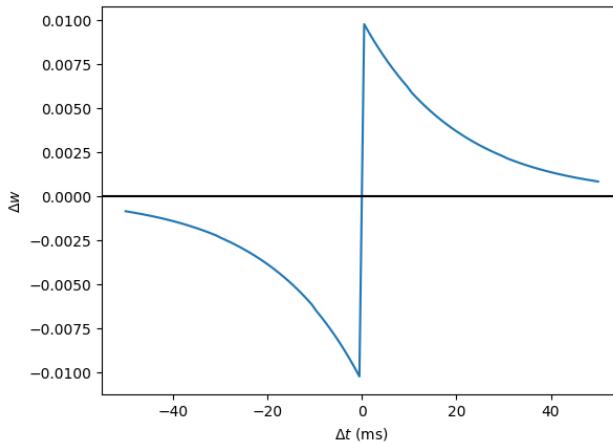


Figure 5 STDP. The change in synaptic weight Δw as a function of the difference in spiking times Δt between the presynaptic and postsynaptic neurons.

STDP

Spike-Timing-Dependent Plasticity (STDP) is a biological phenomenon observed in the brain that underpins one of the fundamental mechanisms for synaptic modification, which in turn affects learning and memory. It describes how the timing of spikes (action potentials) between neurons influences the strength of connections (synapses) between them. Essentially, STDP is a rule by which synapses are strengthened or weakened based on the precise timing of neuronal firing.

In the simplest terms, STDP strengthens synapses if a presynaptic neuron's spike precedes the postsynaptic neuron's spike (pre-before-post), and weakens them if the postsynaptic spike occurs first (post-before-pre). This mechanism is often summarized by the phrase “neurons that fire together, wire together”, indicating that neuronal connections become stronger when the neurons are active in close temporal proximity.

The effect of STDP depends on the time interval between the spikes of the pre- and postsynaptic neurons. Typically, synaptic strength is only significantly affected if the timing difference between the spikes is within a certain window, usually on the order of tens of milliseconds. The amount of synaptic modification decreases as the interval between the spikes increases.

Hebbian STDP is characterized by synaptic strengthening when the presynaptic neuron fires just before the postsynaptic neuron. It is often associated with learning and memory formation.

To set up a simple model to explore Spike-Timing-Dependent Plasticity (STDP), a mechanism where the synaptic strengths between neurons are adjusted based on the relative timing of their spikes is implemented.

Variables ‘taupre’ and ‘taupost’ are set to define the time constants for the presynaptic and postsynaptic traces, respectively, each of 20 milliseconds (ms). ‘Apre’ and ‘Apost’ define the magnitude of spike-triggered increases or decreases in synaptic strength, where ‘Apost’ is adjusted to be slightly more potent than ‘Apre’ (but negative, indicating a decrease in synaptic strength).

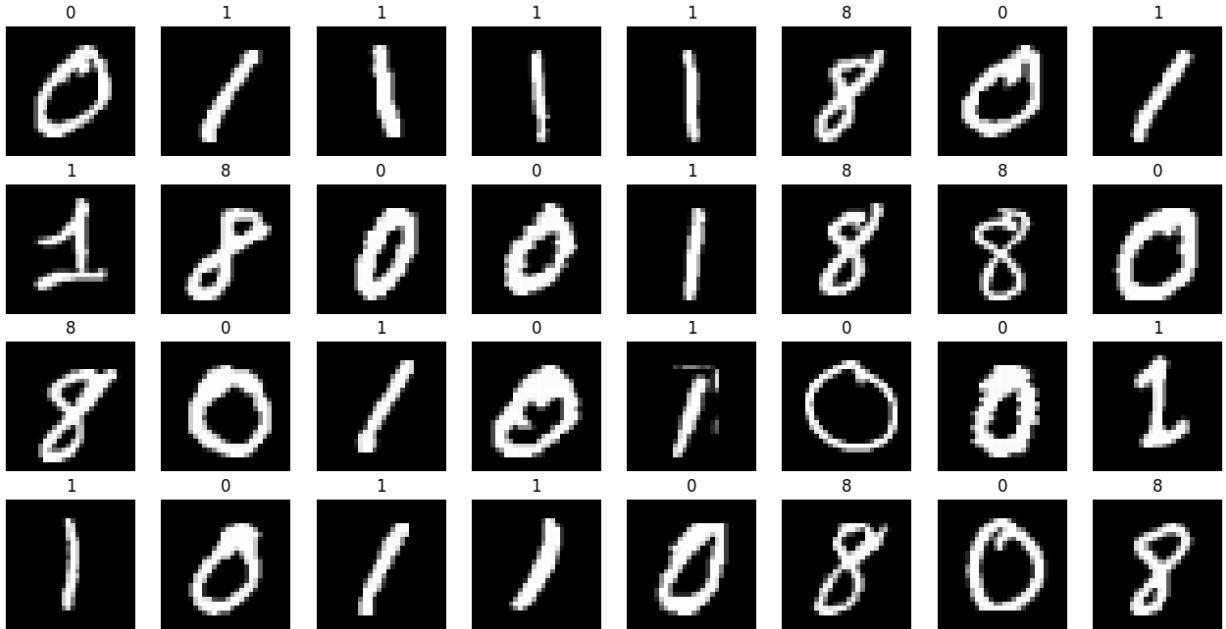


Figure 6 MNIST dataset used for learning.

Two neuron groups, 'G' and 'H', are created using 'NeuronGroup'. 'G' represents presynaptic neurons with a linearly increasing spiking time from 0 to 'tmax'. In contrast, 'H' represents two postsynaptic neurons with spiking times decreasing linearly from 'tmax' to 0, effectively varying the difference in spiking times between pre- and postsynaptic neurons from '-tmax' to '+tmax'.

The 'Synapses' object 'S' is created to link every presynaptic neuron in 'G' to every postsynaptic neuron in 'H'. It includes variables for synaptic weight ('w'), presynaptic activity trace ('apre'), and postsynaptic activity trace ('apost'), with dynamics defined by the equations provided. These equations ensure that 'apre' and 'apost' decay over time according to 'taupre' and 'taupost'.

On a presynaptic spike ('on_pre'), 'Apre' is added to 'apre', and the synaptic weight 'w' is increased based on 'apost'. Conversely, on a postsynaptic spike ('on_post'), 'Apost' is added to 'apost', and 'w' is increased based on 'apre'. This realizes the STDP update rule where the change in synaptic weight depends on the relative timing of pre- and post-spikes.

To make sure the model is correct, a plot is generated showing the change in synaptic weight ' Δw ' as a function of the difference in spiking times ' Δt ' between the presynaptic and postsynaptic neurons. The line 'axhline' adds a horizontal line at ' $\Delta w = 0$ ' for reference (**Figure 5**).

Now that the model works, we proceed to the second task.

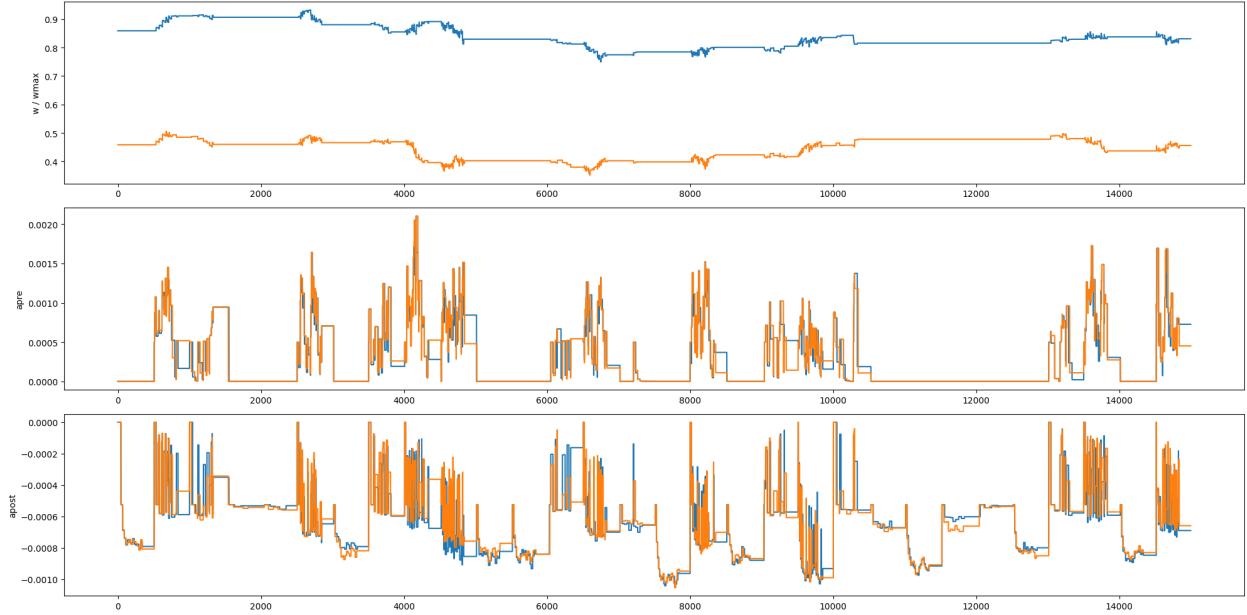


Figure 7 Synapse dynamics. The top panel shows the output neurons weights over time. The middle panel displays the presynaptic trace and the bottom panel displays the postsynaptic trace.

Task 2: Designing a Fully Connected Spiking Neural Network

For convenience MNIST dataset has been used for learning purposes (**Figure 6**). The STDP model ('stdp') includes variables for synaptic weight ('w'), and equations for the evolution of 'Apre' and 'Apost' over time, based on their respective time constants. time constants for the decay of presynaptic (Apre) and postsynaptic (Apost) traces are set to be equal, indicating symmetric decay rates for pre- and post-synaptic activity.

The 'pre' and 'post' strings define the synaptic weight update rules on presynaptic and postsynaptic activities, respectively. When a presynaptic neuron fires (pre-activity), synaptic efficacy is increased by a factor of 'dApre', and similarly for post-activity with 'dApost'. The weight updates are constrained within the range '[0, gmax]' by the 'clip' function to prevent runaway growth or negative weights.

PoissonGroup are used for input signals; this group represents a group of neurons generating spikes following a Poisson process, which typically serves as an input layer in SNNs. The rate of each neuron in the PoissonGroup is initially set to zero.

The output groups are excitatory neurons which are defined by a set of differential equations that describe the dynamics of membrane potential ('v'), excitatory ('ge'), and inhibitory ('gi') conductances. These dynamics incorporate the effects of synaptic inputs ('ge', 'gi'), resting potential, and the leaky integrate-and-fire model parameters.

Neurons in this group have a spiking mechanism defined by a threshold condition ('v>v_thresh_e'), a refractory period of '5ms', and a 'reset' mechanism that sets the membrane potential to 'v_reset_e' upon spiking. The initial membrane potential for each neuron is set to a value below the resting potential ('v_rest_e - 20.*mV') to allow for some variability in the initial conditions.

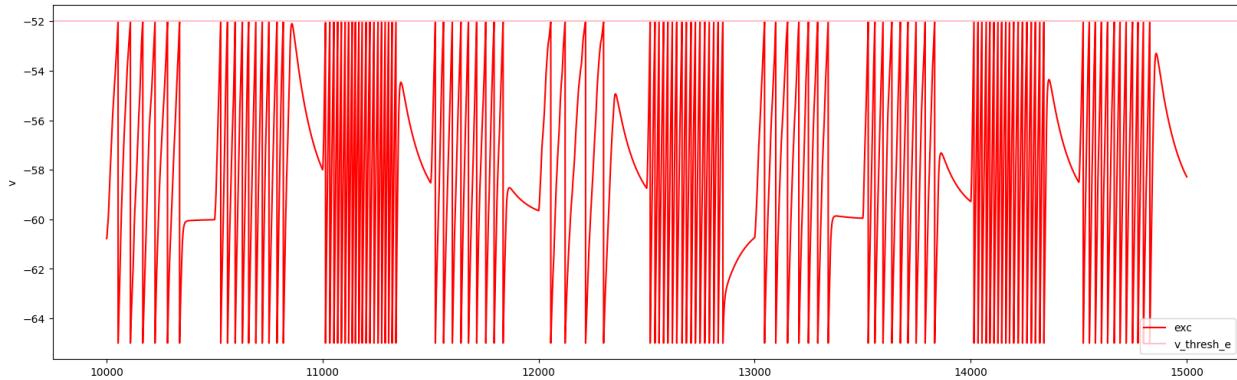


Figure 8 The changes in the membrane potential (v) of output neuron over time.

Synapses connects the Poisson input group ('PG') with the excitatory neuron group ('EG') with STDP rules ('stdp', 'pre', 'post') for dynamic weight adjustment. This configuration enables the network to learn from the temporal patterns of spikes between the input and output neurons through the adjustment of synaptic weights based on the relative timing of presynaptic and postsynaptic spikes.

Through the dynamics of STDP, the network adjusts the synaptic weights ('w') in response to spiking activity, which enables it to learn temporal patterns in the input spike trains. The network's design, integrating Poisson inputs with excitatory neurons and plastic synapses, makes it suitable for tasks like pattern recognition and classification in a spatio-temporal context, leveraging the temporal dynamics of spikes for computation.

Training procedure

Our network has been trained using the numbers 0, 1, and 8. Furthermore, two training schemes have been incorporated. In the first scheme, entire images are presented to the input neurons in each simulation, and their results such as accuracy and the cross-correlation matrix are employed.

Training with the entire image

As it can be seen in **Figure 7**, The top panel shows the output neuron weights over time, which start around 0.7 and fluctuate as learning progresses through the simulation. The middle panel displays the presynaptic trace, which seems to exhibit sharp spikes at various time points. The presynaptic trace represents the recent spike history of the presynaptic neurons and influences synaptic plasticity. The bottom panel shows the postsynaptic trace, which also has spikes but appears smoother compared to the presynaptic trace. The postsynaptic trace tracks the recent activity of the postsynaptic neuron and also impacts synaptic weight updates.

Regarding STDP, this unsupervised learning rule modifies synaptic strengths based on the relative timing of pre- and postsynaptic spikes. The changes in output weights, along with the pre-

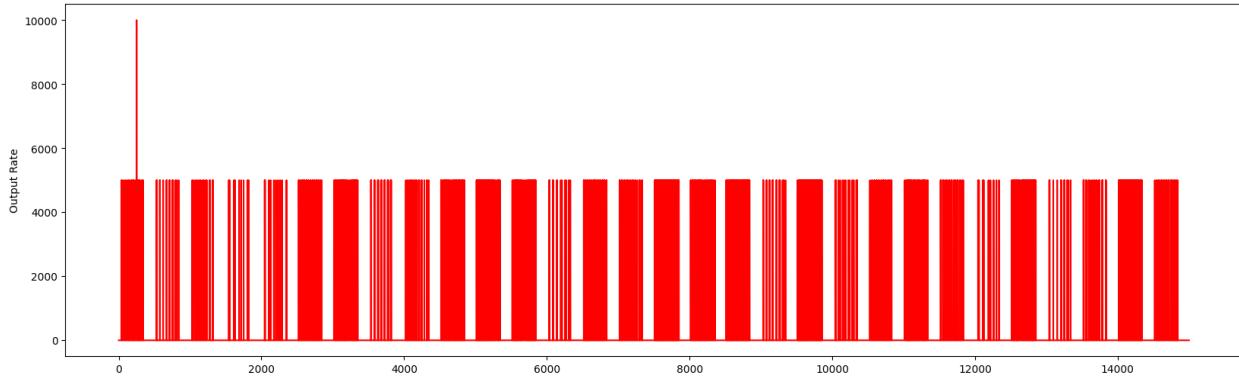


Figure 10 Output population firing rate.

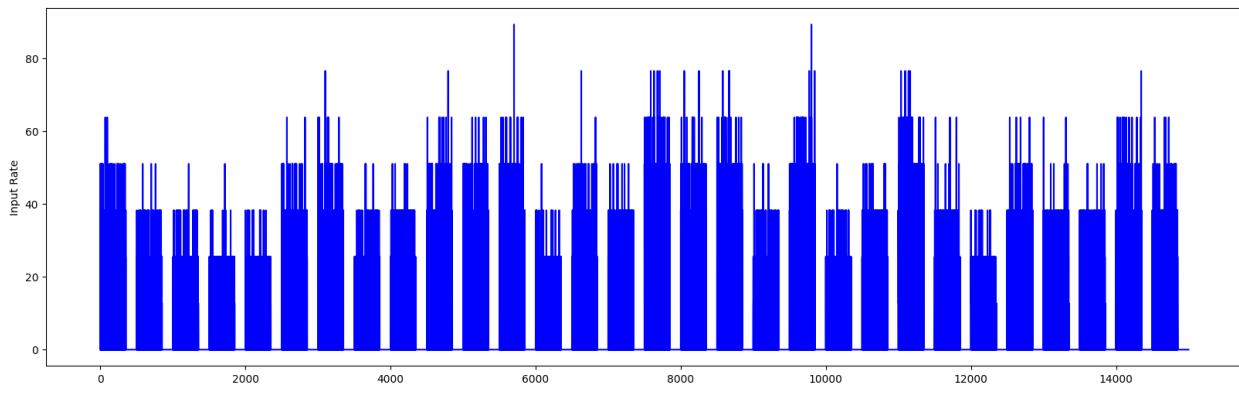


Figure 9 Input population firing rate.

and postsynaptic traces, suggest that STDP is being used to learn spike patterns and adjust the connectivity of the spiking neural network during this simulation.

The fluctuations and spike patterns in the three panels illustrate the dynamics of how STDP shapes the output neuron's weights and responses over time as it learns from the spiking inputs it receives.

Figure 8 plots the changes in the membrane potential (v) of output neuron over time, specifically in the context of spike-timing dependent plasticity (STDP) learning. The plot exhibits a pattern of rapid depolarizations (increases in membrane potential towards 0 mV) followed by repolarizations (decreases back to around -64 mV). These sudden upward deflections in the membrane potential represent action potentials or “spikes” generated by the neuron as it is seen in **Figure 12**.

Figure 9 shows the output neuron population firing rate over time. This plot has a relatively constant, high firing rate around 8000-9000 spikes per time unit with some small fluctuations. On the other hand, **Figure 10** represents the input neuron population firing rate over the same time period. This plot exhibits a strikingly different pattern; the firing rate fluctuates significantly between around 20 to 70 spikes per time unit in an irregular, bursty manner.

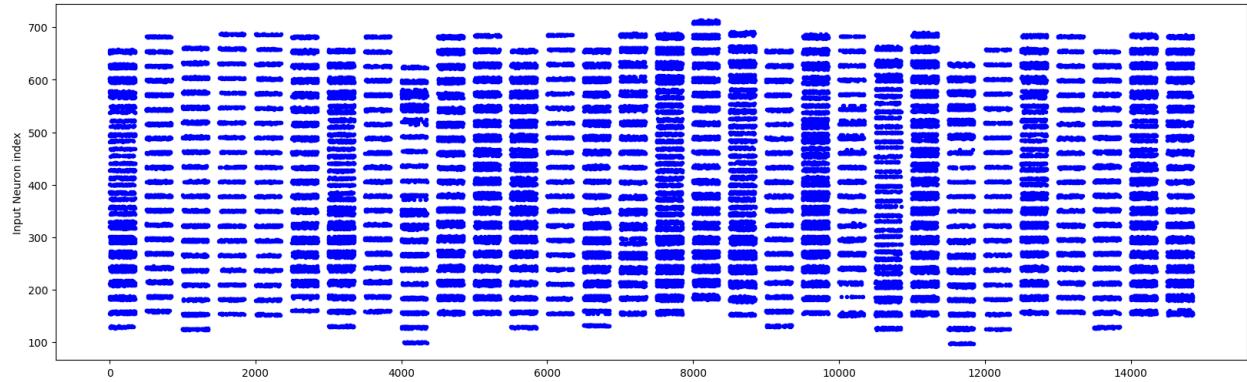


Figure 12 Input neurons raster plot.

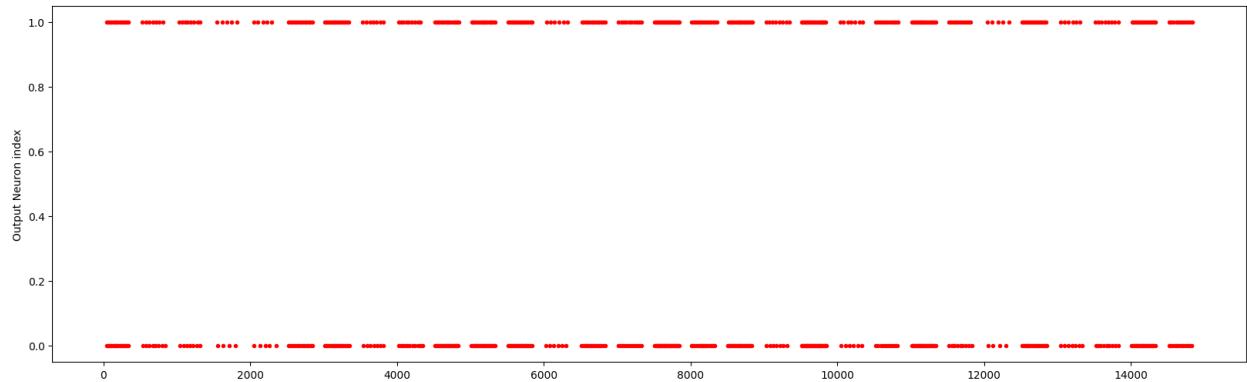


Figure 11 Output neurons raster plot.

The difference in patterns and rates between the output and input neuron populations, as depicted in **Figures 9 and 10**, can be attributed to several factors related to network structure, the nature of the input, and how spiking neural networks (SNNs) process information.

1. The input neuron group comprises 784 neurons, while there are only 2 output neurons. This significant difference in neuron counts naturally leads to different aggregate firing rates. Given that each input neuron has the potential to influence the firing rate of the output neurons through synaptic connections, the cumulative effect of many input neurons on a much smaller number of output neurons can amplify the observed output firing rate.
2. Input neurons typically receive external signals and convert them into patterns of spikes. These neurons are designed to react to varying inputs, which can lead to the irregular, bursty firing patterns observed. The fluctuating firing rates between 20 to 70 spikes per time unit reflect the input neurons' role in responding to a diverse range of stimuli with varying levels of intensity and frequency. Output neurons, however, are meant to consolidate information from the network and produce a more uniform response that represents the network's decision or output state. This can result in a relatively constant and high firing rate, as the output neurons aggregate and respond to the cumulative activity received from the network.
3. Given the STDP rule's role in adjusting synaptic strengths based on the timing of spikes, the network learns to generate a specific pattern of activity at its output neurons in response to the input signals. The learning process could lead to a scenario where the output neurons

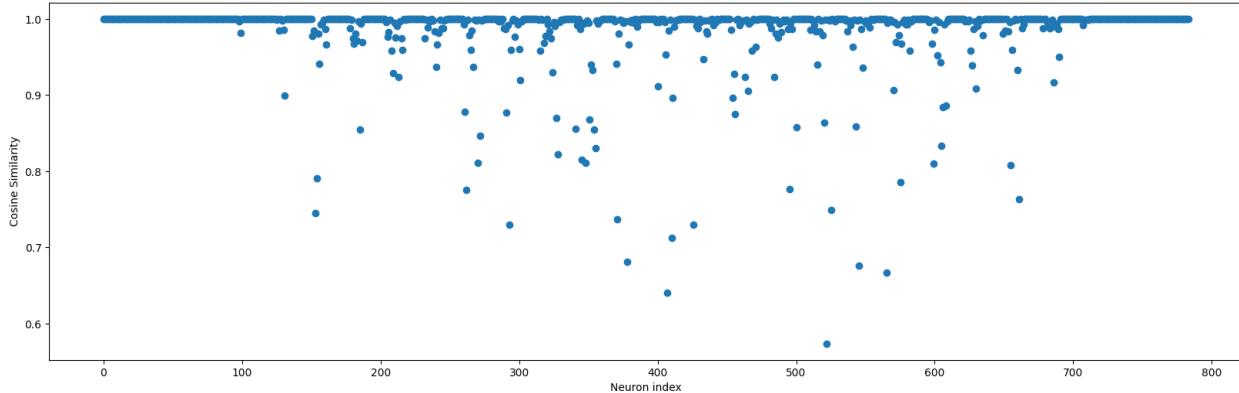


Figure 13 Cosine Similarity STDP learning.

maintain a high and relatively stable firing rate as the network stabilizes and optimizes its internal synaptic connections for the classification task.

4. Output neurons are typically designed to integrate inputs from many neurons before reaching their firing threshold. Due to the large number of input neurons (784), even if each contributes a small effect, the cumulative effect can push the output neurons to fire more consistently and at higher rates. The dynamics of integration and thresholding inherently lead to different activity patterns compared to the variable, directly stimulated input layer.

Comparing the **Figure 11** and **12** side-by-side reveals that the neural network is transforming the dense, distributed input spiking into a very sparse, binary-like output code with spike bursts representing specific patterns. This sparse coding could be the network's way of efficiently representing and encoding the complex, high-dimensional input patterns using few active output units at any point in time, essentially extracting the most salient features. The short spike bursts may also indicate that the output neurons are selectively responding to specific spatio-temporal patterns embedded within the broader, noisy input spike stream through their recurrent dynamics.

Figure 13 is a scatter plot showing the cosine similarity values between pairs of neurons' weights. *Cosine similarity* is a measure of similarity between two non-zero vectors, calculated as the cosine of the angle between them. It ranges from -1 (opposite direction) to 1 (same direction), with 0 indicating the vectors are orthogonal.

In the context of neural networks, cosine similarity can be used to quantify how similar the activation patterns or weight vectors are between pairs of neurons. As we can observe, the cosine similarities range from around 0.6 to 1.0, indicating a high degree of similarity between most neuron pairs. There is also a dense clustering of points around cosine similarity values of 0.9-1.0, suggesting many neuron pairs have very high pattern similarity and only a small number of points fall below 0.8 cosine similarity, representing relatively fewer neuron pairs with more distinct activation patterns.

The predominance of high cosine similarities indicates that within this neural population, most neurons tend to have highly overlapping preferred activation patterns or weight vectors pointing in very similar directions in the high-dimensional space.

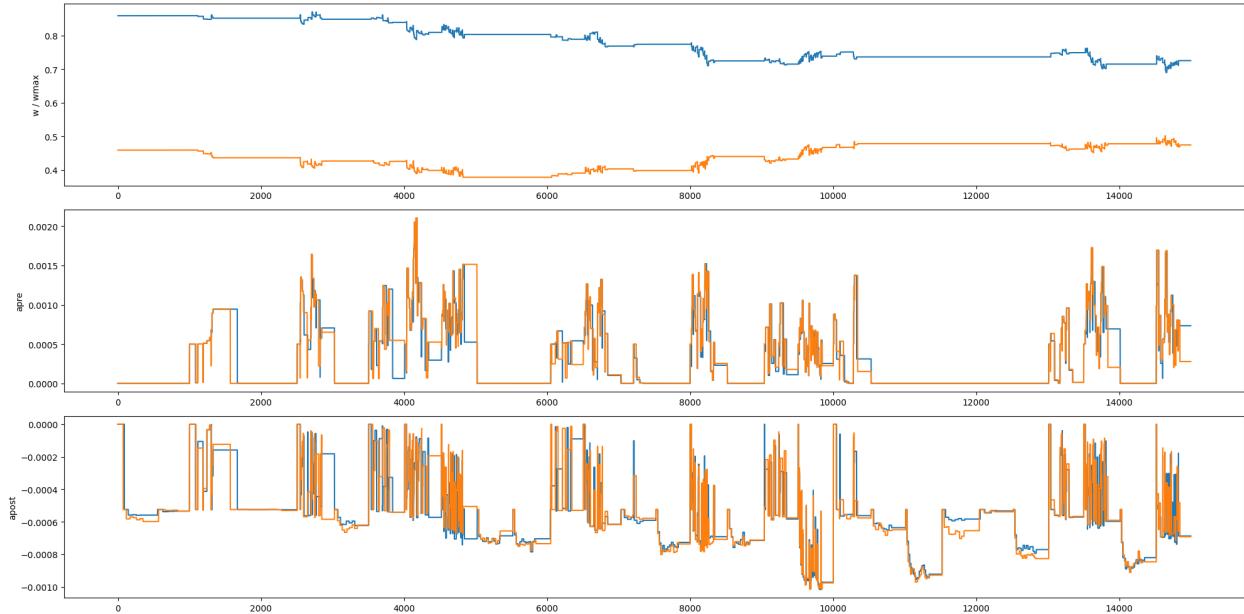


Figure 15 Synapse dynamics. The top panel shows the output neurons weights over time. The middle panel displays the presynaptic trace and the bottom panel displays the postsynaptic trace. All trained with STDP on intersecting images

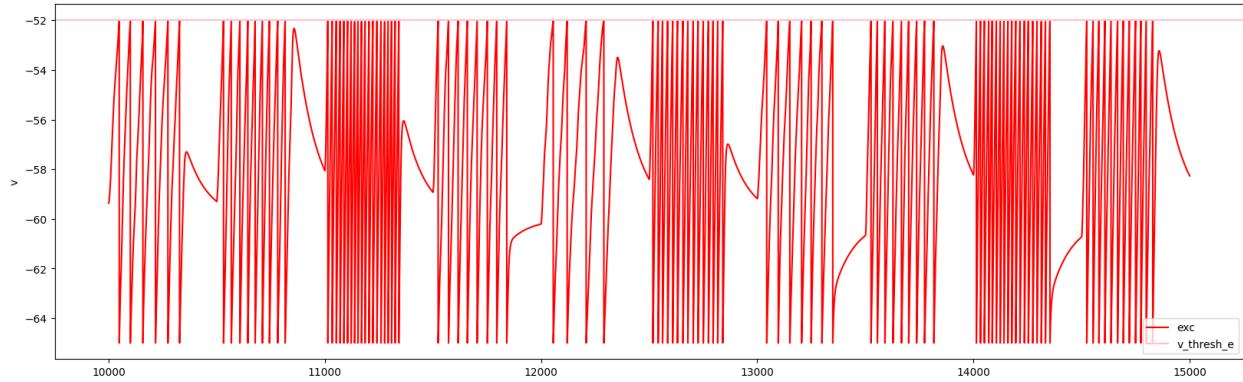


Figure 14 membrane potential on training with intersecting images with STDP.

Classification result

Evaluation of the network over 500 training images and 100 testing images indicates that the classification accuracy can achieve **0.81** during a 3500ms simulation period.

Training with intersecting image

The fluctuations and spike patterns in the three panels of **Figure 14** and **15** illustrate the dynamics of how STDP shapes the output neuron's weights and responses over time as it learns from the spiking inputs it receives. Comparing **Figure 7** and **8** with **Figure 14** and **15**.

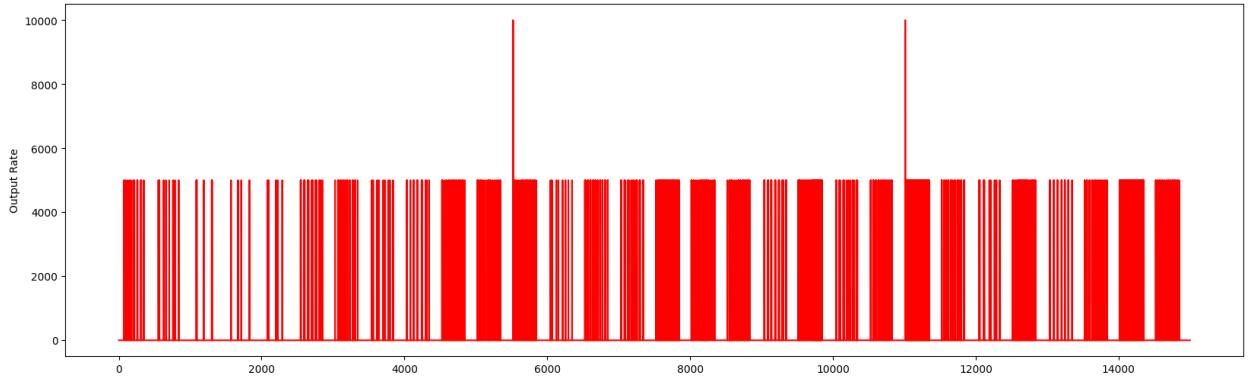


Figure 17 Output population rate with STDP learning and intersecting images.

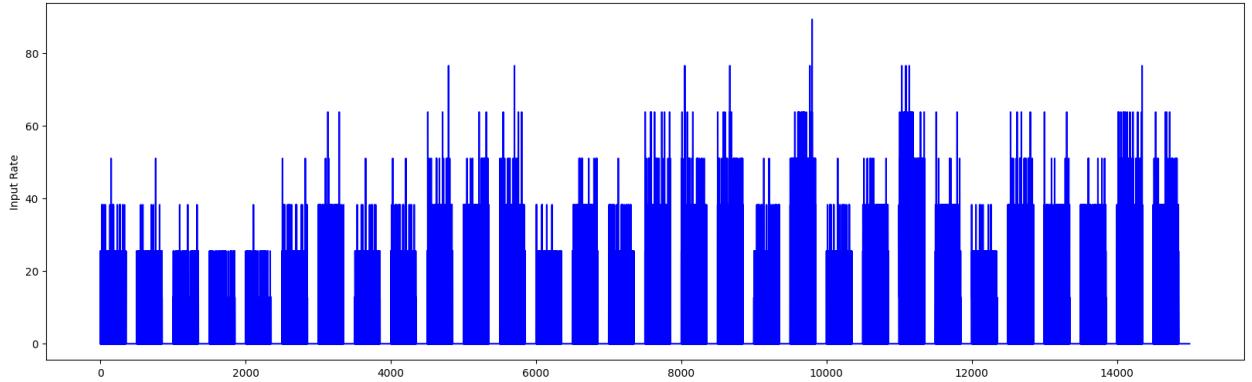


Figure 16 Input population rate with STDP learning and intersecting images..

Also we can see difference mostly in the first half of population firing rate plots from the previous training scheme (**Figure 9 and 10**) and the current scheme (**Figure 16 and 17**). This is because the images are not the same and are only part of what should be seen by the inputs.

This is clearly represented in **Figure 18** where the spiking neurons, random choice of the the pattern activity, and intersecting nature of the input pattern over simulation is seen. this causes a different spiking pattern of output neuron (**Figure 19**) from what we have seen previously in **Figure 12**.

Based on the observations from **Figure 20**, the cosine similarity within this training methodology generally approaches 1, with fewer values clustered around zero. This distribution pattern underscores the enhanced efficacy of the current model relative to the previous one. It seems like gradually presenting the whole image to the input neurons works better with Spike-Timing-Dependent Plasticity (STDP) compared to showing the whole image all at once. There could be several reasons supporting this observation:

1. *Temporal Learning Enhancement:* STDP is a temporal learning rule that adjusts synaptic weights based on the precise timing of spikes between pre- and postsynaptic neurons. By gradually presenting the image, spikes are spread out over time, which can enhance the ability of STDP to capture and reinforce meaningful temporal patterns of activity that represent the image. This temporal aspect of learning is less pronounced when the whole

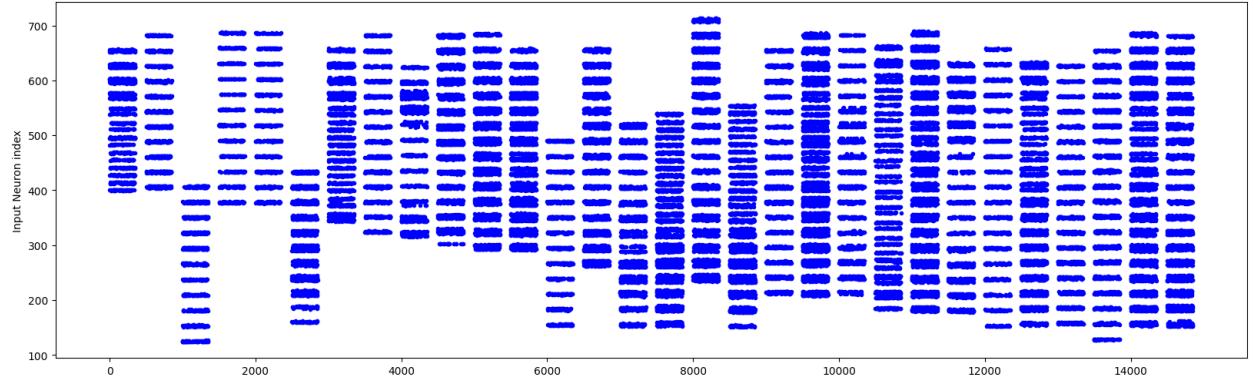


Figure 19 Input neurons raster plot for STDP learning with intersecting images.

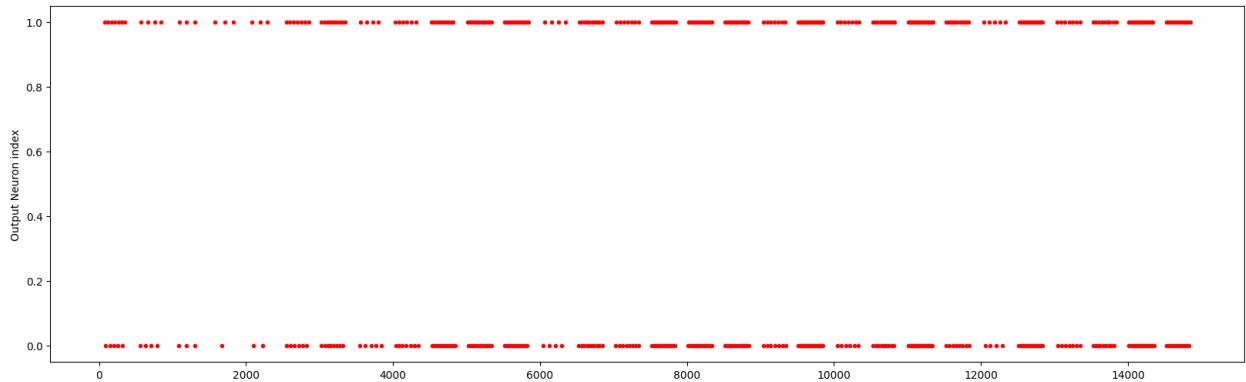


Figure 18 Output neurons raster plot for STDP learning with intersecting images.

image is presented at once, as the synchronous firing may not provide as rich temporal dynamics for STDP to exploit.

2. *Focused Attention and Noise Reduction:* Gradual presentation can mimic a sort of attention mechanism, focusing the network's processing resources on different parts of the image over time. This can help in reducing the noise from less relevant parts of the image by emphasizing the learning of features in a sequential manner. It enables the network to build a more complex and nuanced representation of the image as it "learns" different parts sequentially.
3. *Increased Spike Discrimination:* By presenting the image gradually, neurons are less likely to become overwhelmed by excessive stimulation, which can happen when the entire image is presented at once. This can prevent neurons from reaching their firing threshold too rapidly and indiscriminately, allowing for a more discriminated response that favors synapses which are genuinely relevant to the task. The increased discrimination can lead to more effective and efficient synaptic weight adjustments by STDP.
4. *Enhanced Feature Extraction:* Gradually presenting the image allows neurons more time to respond to specific features, enhancing the network's ability to extract and learn about these features over time. STDP can exploit this prolonged engagement with specific image features to strengthen synapses associated with the most relevant features for classification purposes, improving overall performance.

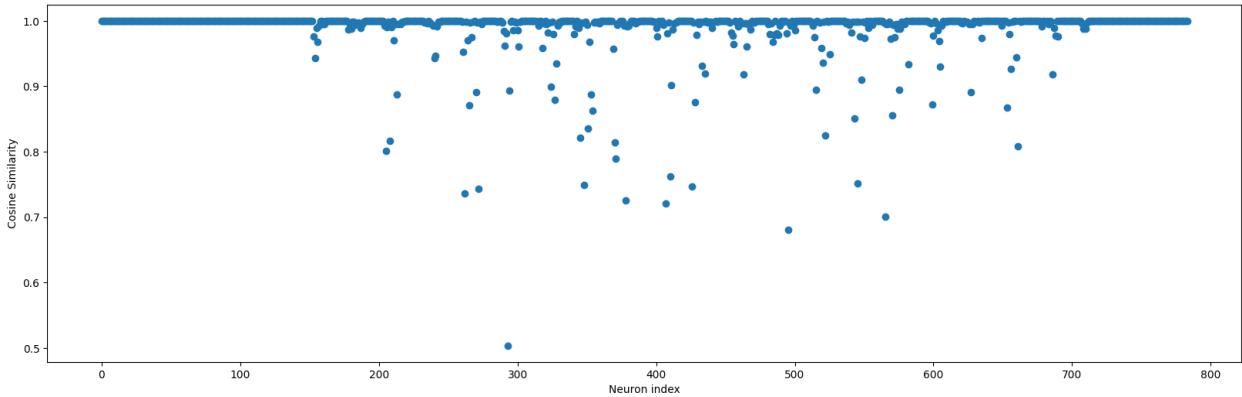


Figure 20 Cosine Similarity STDP learning with intersecting image.

Classification result

Evaluation of the network over 500 training images and 100 testing images indicates that the classification accuracy can achieve **0.84** during a 3500ms simulation period which is a slightly better result than the previous scheme.

Task 3: Adding dead neurons (STDP)

In this task, we introduced an additional neuron to both the input and output layers, designed not to spike. This setup ensures that, throughout each timestep of the simulation, the membrane potential of the output neuron remains at its resting state (**Figure 22**). Regarding the input layer, as we construct the timed array for firing rates based on a Poisson distribution, we include a neuron with a firing rate set to zero. This effectively prevents it from spiking, rendering it inactive.

Figure 21 illustrates the changes in synapse weights, alongside postsynaptic and presynaptic traces. The weight of the newly added neuron (represented by the green line in the top panel) remains constant over time, holding its initial value calculated by `'rand() * max_w'`. This stationary weight demonstrates that, unlike the other two output neurons, it does not participate in the learning process, where changes in their weights reflect adaptation and learning.

In **Figure 21** middle panel, representing the presynaptic trace from the input layer, we observe fluctuations in all three synapses. This variability is due to the fact that a presynaptic neuron's weight adjustments are influenced by all three output neurons. Hence, even if one output neuron does not impact the presynaptic change, the other two ensure variation and adjustment in synaptic strength.

The bottom panel, showcasing the postsynaptic trace, again highlights the distinct behavior of the third neuron, which maintains a trace value of *zero*, indicating no postsynaptic activity. In contrast, the other two neurons show fluctuations, indicating active participation in the network's synaptic interactions.

Postsynaptic traces are mostly negative, and presynaptic traces are positive due to the different roles they play in neural signal transmission and the nature of their electrical properties:

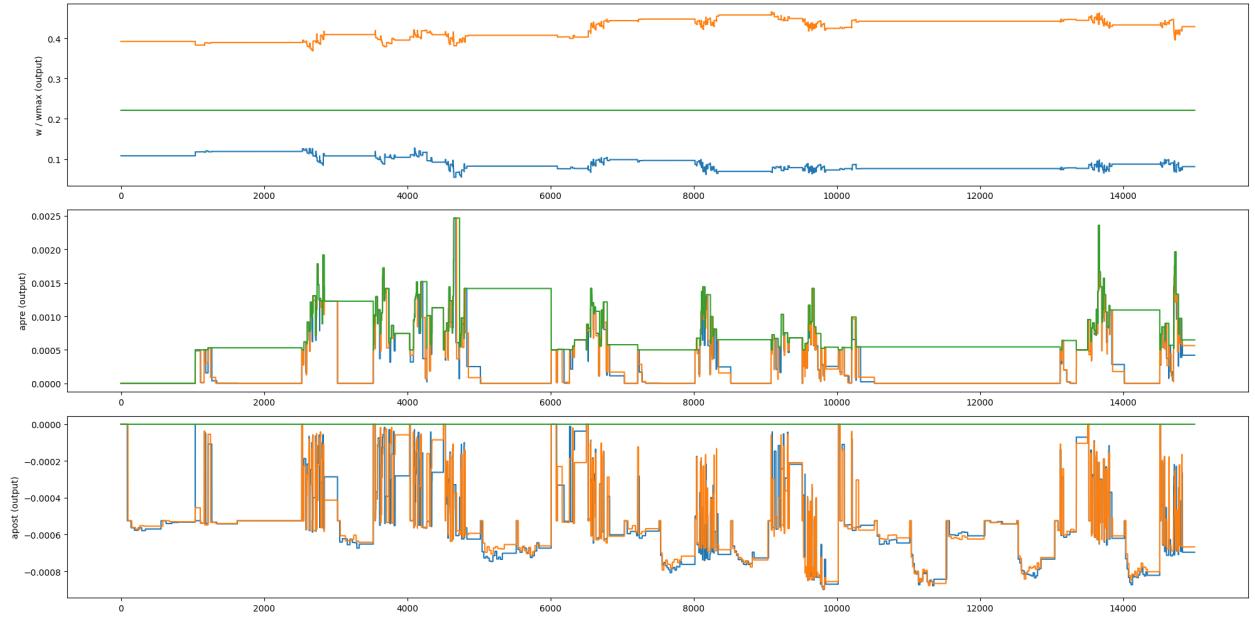


Figure 22 Synapse dynamics. The top panel shows the output neurons weights over time. The middle panel displays the presynaptic trace and the bottom panel displays the postsynaptic trace. Two inactive neurons added to input and output layers.

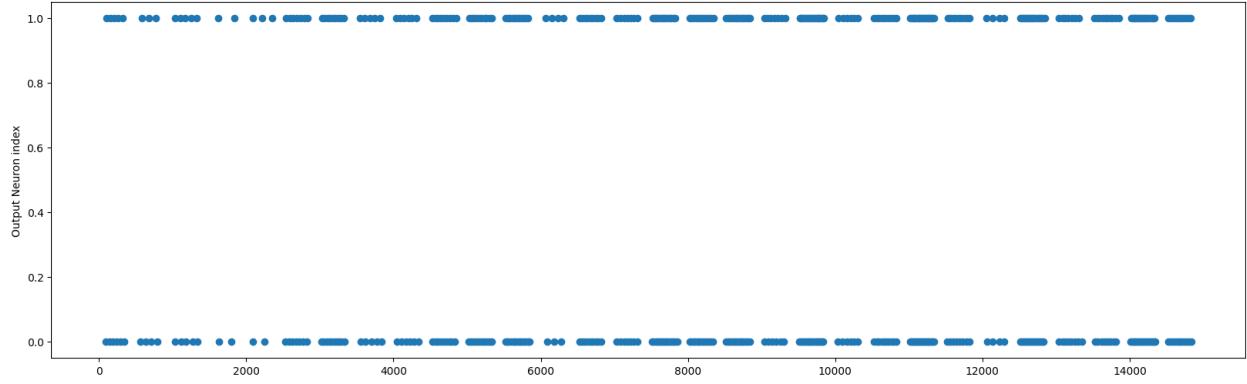


Figure 21 Raster plot for 3 output neurons. Patterns learned with STDP and intersecting images.

1. Presynaptic Traces (Positive): The presynaptic neuron is responsible for sending signals. It generates action potentials, which are positive shifts in voltage, leading to the release of neurotransmitters. These positive shifts or spikes in electrical activity are hence represented by positive traces, indicating the occurrence of an action potential or the presynaptic activity leading up to neurotransmitter release.
2. Postsynaptic Traces (Negative): The postsynaptic neuron receives the signal. The binding of neurotransmitters to the postsynaptic receptors can lead to either depolarization or hyperpolarization, depending on the type of neurotransmitter and receptor. Inhibitory synaptic inputs often result in hyperpolarization, making the inside of the neuron more negative, which can be interpreted as negative traces in the context you're referring to. This negative trace reflects the neuron's inhibitory response, although excitatory responses can also occur but are not the focus of your question.

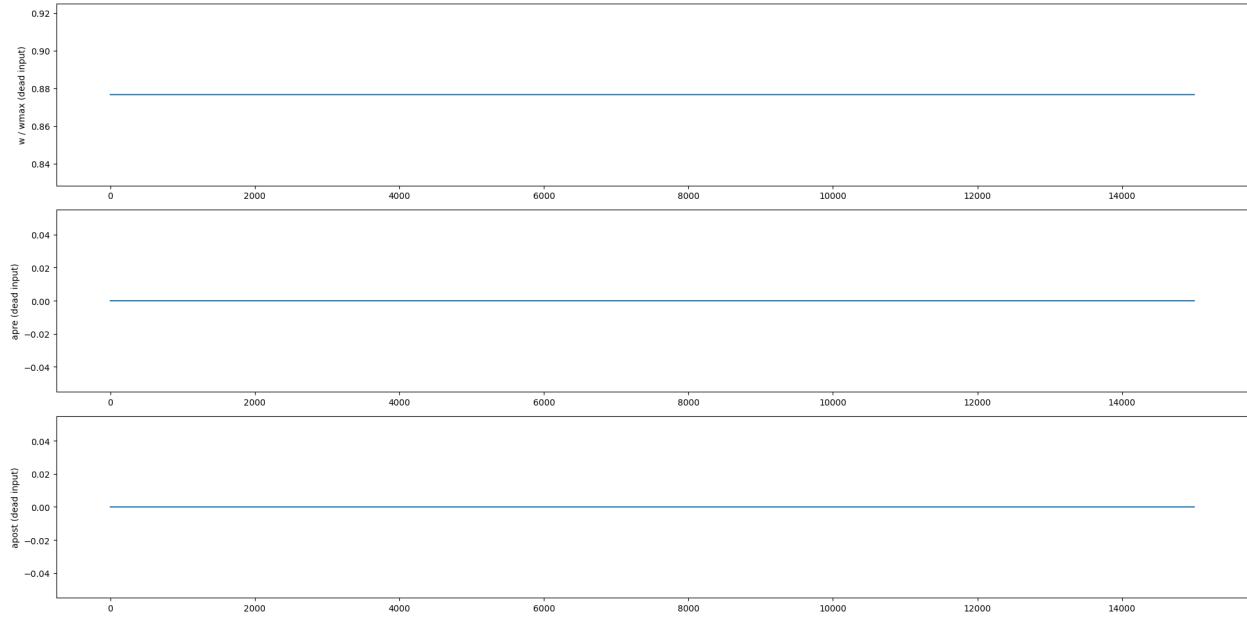


Figure 23 Synapse connecting the two inactive neurons in input and output layers.

It's important to note that the negativity or positivity of postsynaptic traces can also be influenced by the type of recording or the specific aspect of postsynaptic activity being measured. For example, in the context of synaptic plasticity and learning, the difference in timing between presynaptic and postsynaptic spikes (e.g., in STDP) is crucial, and the way these are graphically represented (positive for presynaptic and negative for postsynaptic) simplifies the understanding of their respective roles in synaptic modification.

As expected, the synapse connecting the two inactive neurons remains unchanged throughout. Its weight retains its initial value. The presynaptic trace consistently shows a value of 0 during training, attributable to the zero firing rate of the input neuron. Similarly, the postsynaptic trace is also zero, reflecting the inactivity of the output neuron (**Figure 23**).

Task 4: Set minimum firing rate (STDP)

In this task, our objective is to establish a minimum firing rate, 'min_fr', for all neurons within the network. This adjustment affects two main groups of neurons, the output neurons (denoted as 'EG') and input neurons (denoted as 'PG'). The process is detailed below:

For each output neuron ('EG'), within a 1-second time window, we tally the spikes that occurred. The firing rate is then determined by dividing the count of spikes by the window's duration. If a neuron's firing rate falls below 'min_fr', we stimulate this neuron by adjusting its membrane potential to a level 10 millivolts (mV) above the threshold potential.

In the case of each input neuron ('PG'), the firing rate is directly retrieved since it follows a Poisson distribution. Should the firing rate dip below 'min_fr', we adjust the neuron's firing rate to 'min_fr' to ensure stimulation.



Figure 24 Synapse dynamics. The top panel shows the output neurons weights over time. The middle panel displays the presynaptic trace and the bottom panel displays the postsynaptic trace. Minimum firing rate applied to all neurons.

As observed in **Figure 24**, the synaptic weight of the third neuron also varies throughout the simulation period, with the postsynaptic trace fluctuating due to the implementation of the minimum firing rate setting. Furthermore, in **Figure 25**, activity is reignited between two previously inactive neurons, demonstrating that the method effectively ensures that no neurons remain inactive. This shows the efficiency of the approach in maintaining neural network activity and preventing inactivity within the network.

Task 5: Reward modulated STDP (R-STDP)

Reward-modulated spike-timing-dependent plasticity (STDP) is an advanced neural learning rule that integrates the concept of reward signals into the traditional STDP mechanism. Traditional STDP is based on the principle that the synaptic strength between neurons is adjusted according to the timing of their action potentials (spikes): If a presynaptic neuron fires shortly before a postsynaptic neuron, the synaptic connection between them is strengthened (potentiation), and if the order is reversed, the connection is weakened (depression). This timing-dependent rule is fundamental for learning and memory formation in the brain.

Reward-modulated STDP takes this a step further by incorporating the effects of a global reward signal, which modulates the degree of synaptic plasticity based on the outcome of an action. This is closely aligned with theories of reinforcement learning, where learning is directed and reinforced by feedback from the environment. The core idea is that synaptic changes through STDP are not only determined by the precise timing of spikes but also by whether the outcome of these spikes (in terms of the organism's behavior) is favorable (rewarding) or not.

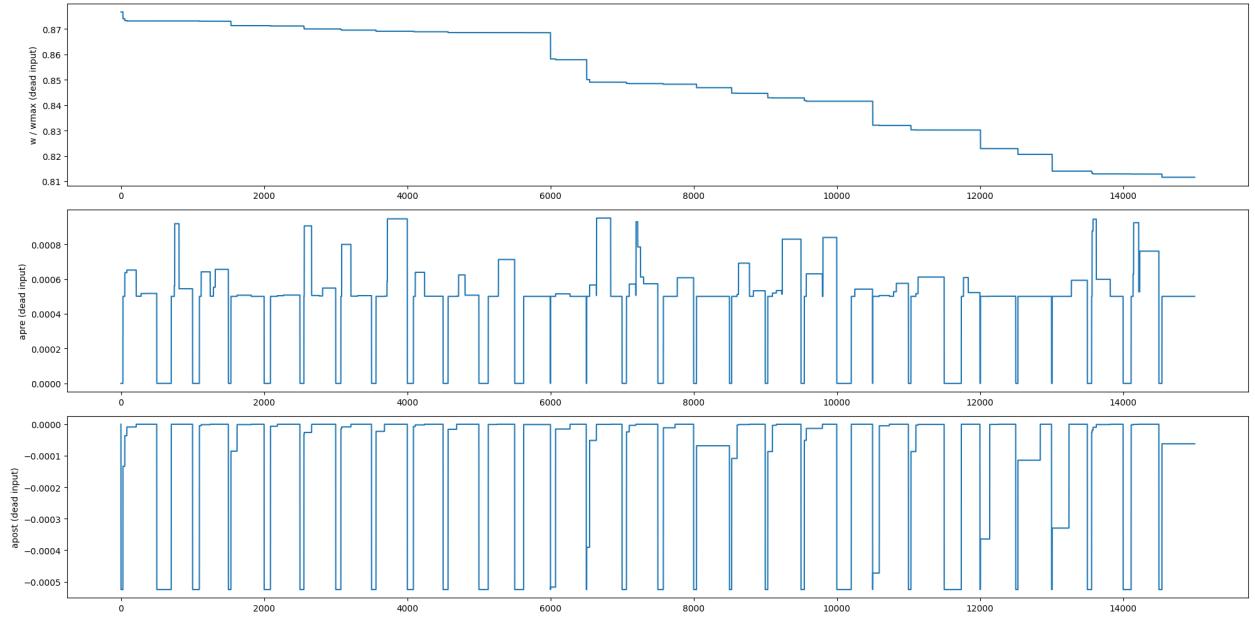


Figure 25 Synapse connecting the two previously inactive neurons in input and output layers.

STTC

The Spike-Time Tiling Coefficient (STTC) is a measure designed to quantify the degree of synchrony or correlation between two spike trains, which are sequences of discrete action potentials recorded from neurons. Unlike traditional correlation measures, the STTC takes into account not only the presence of spikes but also their timing, making it highly suited for analyzing temporal patterns in neural activity. The formulation of STTC accounts for temporal precision in neural interactions, emphasizing spike timing over merely spike occurrence.

STTC is calculated as follows:

$$SSTC = \frac{1}{2} \left(\frac{PA - TA}{1 - PA * TA} + \frac{PB - TB}{1 - PB * TB} \right)$$

PA and **PB** are the proportions of spikes in one train that have at least one matching spike in the other train within a predefined temporal window (Δt). Mathematically, **PA** for train A with respect to train B is defined as the number of spikes in A that are within Δt of any spike in B, divided by the total number of spikes in A, and similarly for **PB**.

TA and **TB** represent the proportion of the total recording time during which spikes in one train could have a matching spike in the other train within the temporal window. Effectively, **TA** and **TB** are measures of the ‘opportunity’ for correlation, accounting for the density of spike opportunities across the recording duration.

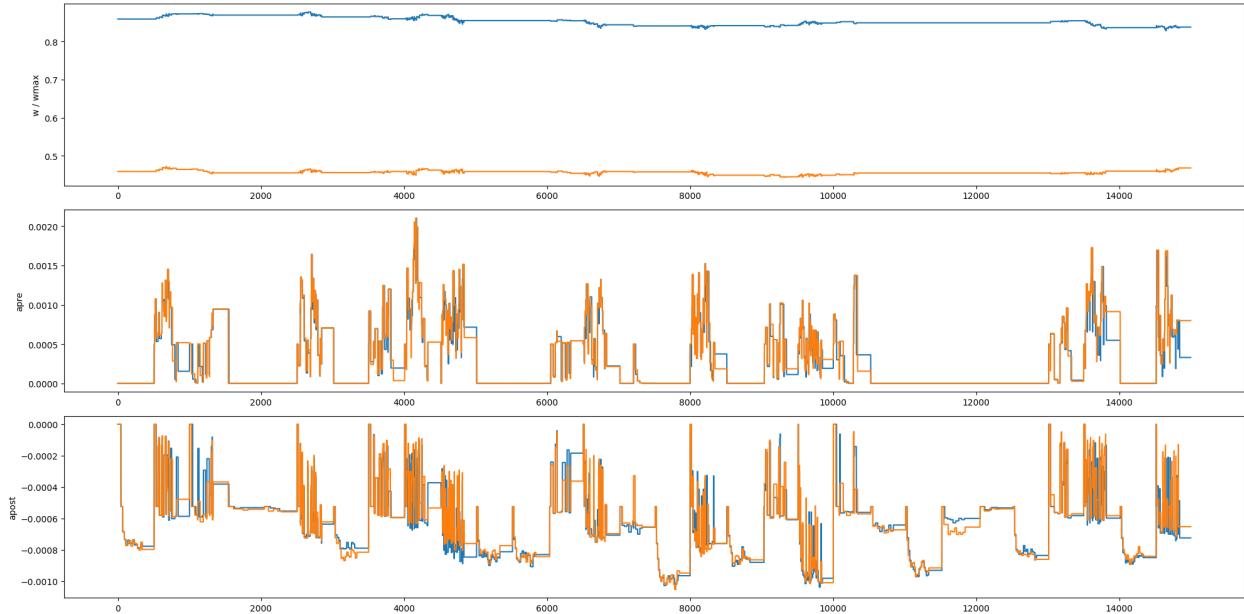


Figure 26 Synapse dynamics with R-STDP learning.

Training with the entire image

In models of neural plasticity, weight fluctuations represent changes in the synaptic strength over time. R-STDP introduces a mechanism where synaptic changes are influenced not just by the precise timing of pre- and postsynaptic spikes, but also by a reward signal. This integration of reward signals into the traditional STDP framework can lead to decreased weight fluctuations in R-STDP compared to traditional STDP models (**Figure 26**).

In traditional STDP, synaptic modifications are solely determined by the relative timing of spikes between the presynaptic and postsynaptic neurons, which could lead to fluctuations as spikes occur randomly over time. R-STDP, however, incorporates an additional layer of modulation based on the outcome of actions (reward signals), providing a goal-oriented bias to the learning process. This means that synaptic adjustments are more systematically oriented towards configurations that lead to rewarding outcomes, potentially reducing the extent of fluctuations due to less randomness in the learning process.

The reward component in R-STDP can help stabilize synaptic weights by reinforcing changes that are associated with positive outcomes and discouraging changes that are not. This reward-driven stabilization means that once synaptic weights are adjusted in a manner that leads to a rewarding outcome, further modifications are selectively reinforced, leading to a decrease in random weight fluctuations observed in standard STDP, where synaptic changes are not influenced by the outcomes of actions.

In our system, the reward is determined by the similarity of spike patterns between input and output, measured by the Spike-Time Tangle Coefficient (STTC), an increase in reward over time suggests that the similarity between these patterns is improving across successive trials or learning episodes (**Figure 27**).

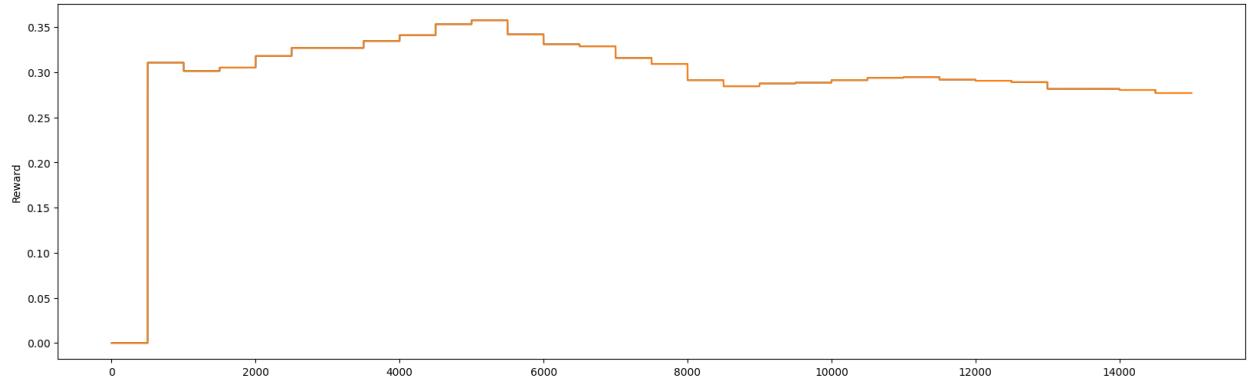


Figure 28 Synaptic reward during training time course.

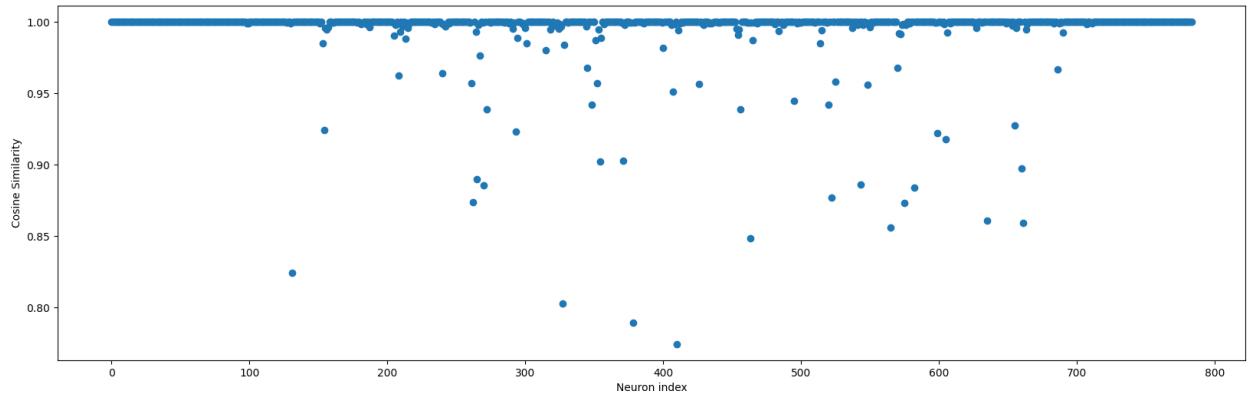


Figure 27 Cosine Similarity RSTDP learning.

Over time, the neurons involved in generating the output pattern may become better tuned to the characteristics of the input pattern. This optimization process is driven by learning mechanisms, which could be facilitated by synaptic plasticity, where the synaptic weights adjust to enhance the congruency between input and output spikes.

Also in our system, the incorporation of reward signals based on STTC similarity can further refine synaptic adjustments. Pathways that contribute to producing output spike patterns similar to the input patterns are selectively reinforced, leading to an increased similarity and thus higher rewards over time. The mechanism might also involve an increased coordination in spike timing between neurons responsible for generating the output pattern. This coordination can be a result of intrinsic network dynamics adjusting to maximize the reward (STTC in this case), leading to a more refined temporal structure of the output spike pattern.

Cosine similarity in the context of R-STDP is slightly closer to 1 compared to STDP (**Figure 28**). suggests a higher degree of alignment or similarity between vectors (e.g., spike trains, neuronal activation patterns, or synaptic weight vectors) in R-STDP models. This outcome can be attributed to several factors inherent to the integration of reward signals into the synaptic plasticity mechanism, emphasizing goal-directed learning and the refinement of neural representations.

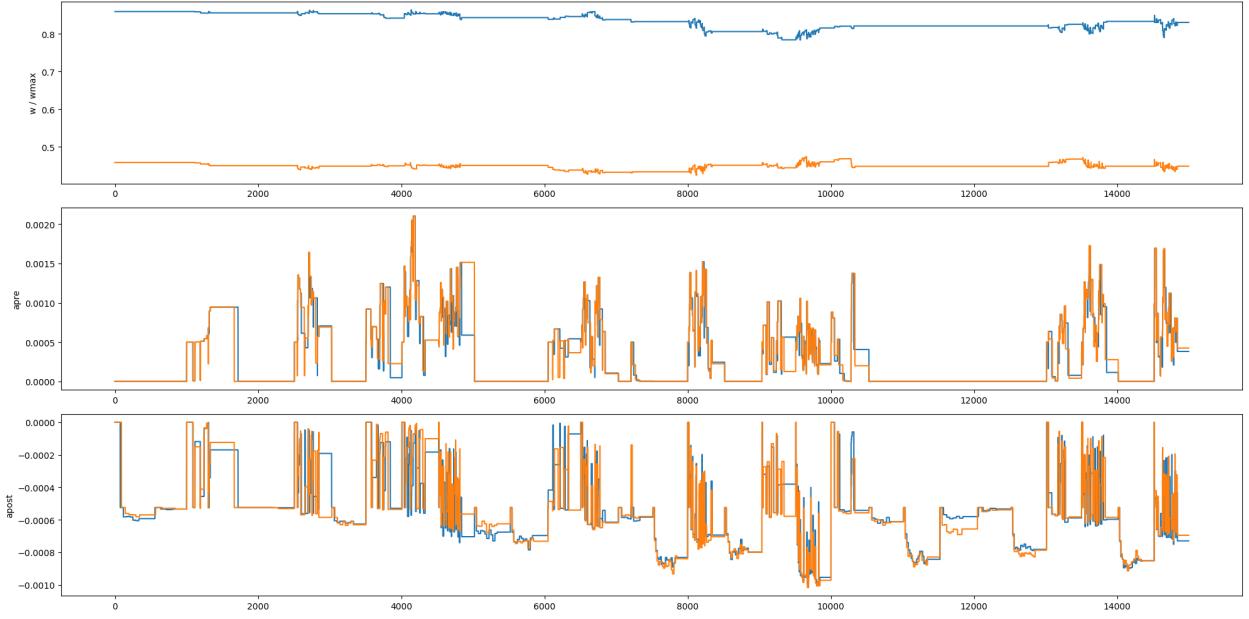


Figure 29 Synapse dynamics with R-STDP learning and intersecting image.

Classification result

Evaluating the network with 100 training and 10 testing images over a 3500ms simulation period shows that it can reach a classification accuracy of **0.86**. This outcome surpasses all previous models, especially notable since the number of training and test images is 5 times larger than in other simulations.

Training with intersecting image

In this scenario, weight fluctuations are smaller than those observed in its corresponding Spike-Timing-Dependent Plasticity (STDP) simulation. The pre- and postsynaptic traces remain unchanged, as modulation of synaptic weight by rewards affects only the weight itself (**Figure 29**).

Towards the simulation's end, it becomes evident that synaptic weight fluctuations are more pronounced when input is presented in intersecting chunks than when the entire image is shown to the input layer at once. This could be attributed to the necessity of Incremental Learning and Integration; presenting input in intersecting chunks forces the neural network to gradually piece together information over time to form a complete image. This integration process requires constant adjustment of synaptic weights with each new piece of information, leading to increased fluctuations. Different chunks may highlight various features or aspects of the image, necessitating ongoing synaptic recalibration by the network to synthesize these elements into a unified representation.

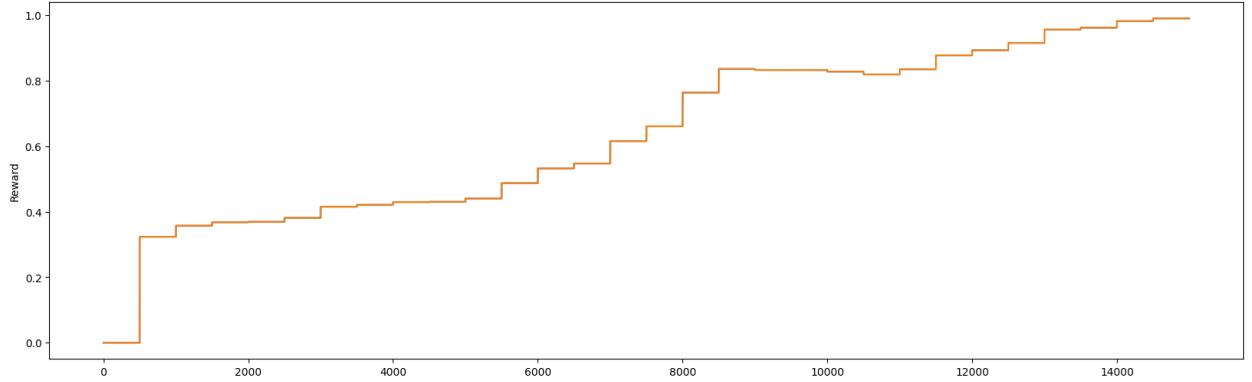


Figure 31 Synaptic reward during training time course with RSTDP learning and intersecting images.

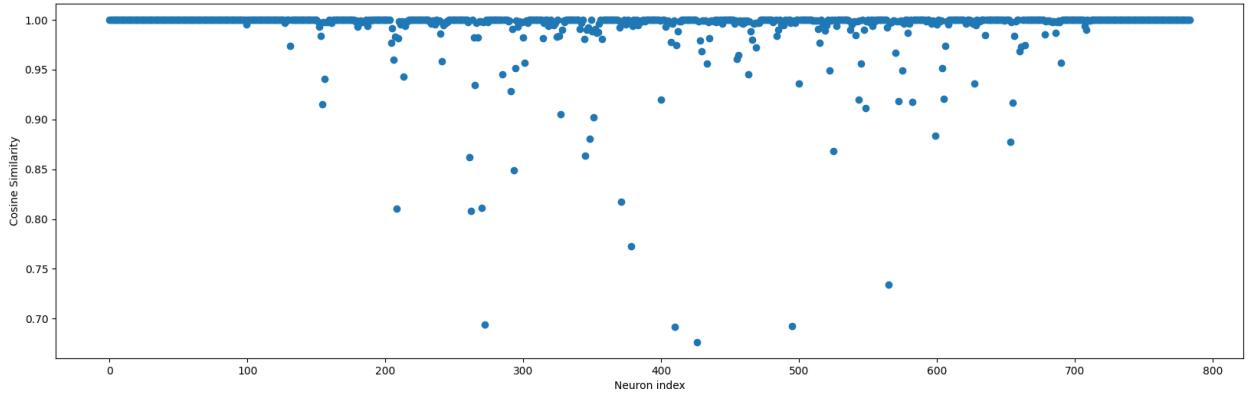


Figure 30 Cosine Similarity RSTDP learning and intersecting images.

We can see an upward trend in the reward signal reaching its maximum of 1 when presenting input in intersecting chunks (**Figure 30**), compared to the weight signal reaching a maximum of 0.35 when the entire image is shown to the input layer at once (**Figure 27**).

Presenting input in intersecting chunks fosters a learning environment where the neural network must engage in more complex spatio-temporal dynamics. This process necessitates the network to not only recognize parts of the input but also understand and anticipate their temporal and spatial relations to other parts. Higher rewards for chunked inputs reflect the network's success in achieving high STTC similarity between the dynamically evolving input and output patterns, a task that demands nuanced temporal coherence and spatial understanding.

In addition, The chunked presentation of inputs compels the network to refine its sensitivity to spatio-temporal patterns, thereby improving its ability to predict successive inputs and their integration into a coherent whole. This enhanced sensitivity and the consequent successful prediction or integration result in higher STTC similarity between input and output, leading to increased reward signals. Essentially, the network is being rewarded for its proficient handling of complex, time-evolving patterns, a task that is more demanding and informative than static pattern recognition.



Figure 32 Synapse dynamics. The top panel shows the output neurons weights over time. The middle panel displays the presynaptic trace and the bottom panel displays the postsynaptic trace. Two inactive neurons added to input and output layers. Learning with RSTDP and intersecting images.

Cosine similarity reveals that when the image is presented in intersecting chunks, input integration with RSTDP results in a slightly higher degree of correlation compared to its use with standard STDP and the prior approach of RSTP, where the image was shown to the input in its entirety (**Figure 31**).

Classification result

Evaluating the network with 50 training and 10 testing images over a 3500ms simulation period shows that it can reach a classification accuracy of **0.9**. This outcome surpasses all previous models, especially notable since the number of training and test images is 10 times larger than in other simulations.

Task 6: Adding dead neurons (RSTDP)

In this task, we introduced an additional neuron to both the input and output layers, designed not to spike. This setup ensures that, throughout each timestep of the simulation, the membrane potential of the output neuron remains at its resting state (**Figure 32**). Regarding the input layer, as we construct the timed array for firing rates based on a Poisson distribution, we include a neuron with a firing rate set to zero. This effectively prevents it from spiking, rendering it inactive.

Figure 32 illustrates the changes in synapse weights, alongside postsynaptic and presynaptic traces. The weight of the newly added neuron (represented by the green line in the top panel) remains constant over time, holding its initial value calculated by `'rand() * max_w'`. This stationary

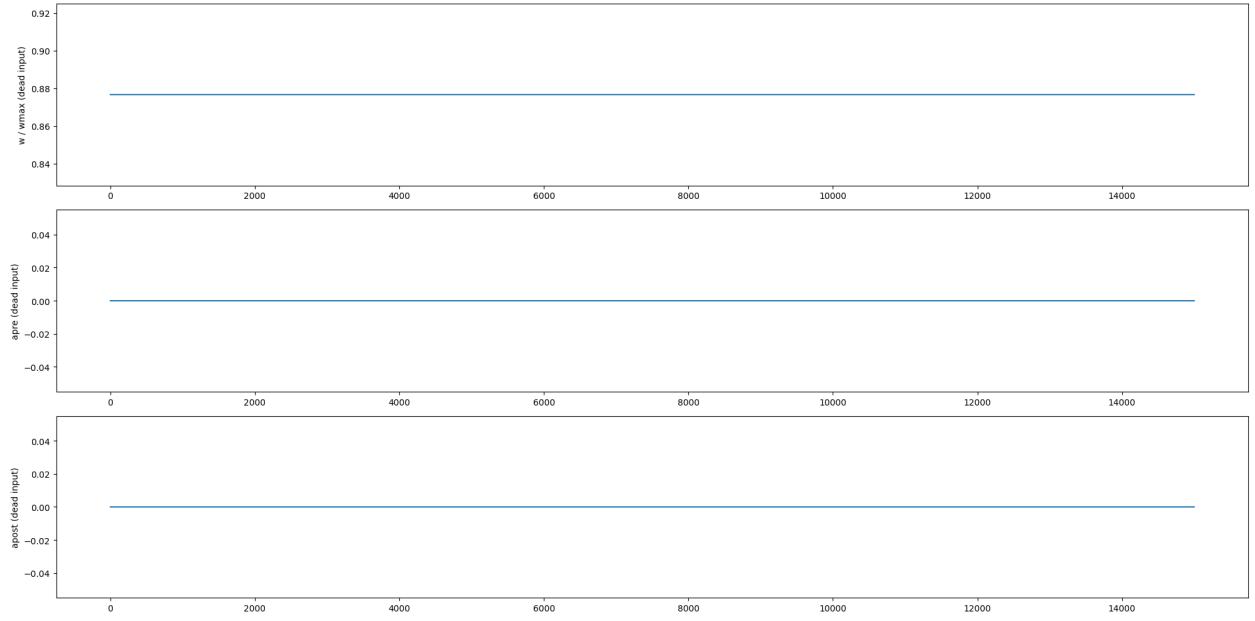


Figure 33 Synapse connecting the two inactive neurons in input and output layers. Learning with RSTDP.

weight demonstrates that, unlike the other two output neurons, it does not participate in the learning process, where changes in their weights reflect adaptation and learning.

In **Figure 32** middle panel, representing the presynaptic trace from the input layer, we observe fluctuations in all three synapses. This variability is due to the fact that a presynaptic neuron's weight adjustments are influenced by all three output neurons. Hence, even if one output neuron does not impact the presynaptic change, the other two ensure variation and adjustment in synaptic strength.

The bottom panel, showcasing the postsynaptic trace, again highlights the distinct behavior of the third neuron, which maintains a trace value of *zero*, indicating no postsynaptic activity. In contrast, the other two neurons show fluctuations, indicating active participation in the network's synaptic interactions.

The postsynaptic and presynaptic traces exhibit consistent patterns across both Figure 32 and Figure 21. The primary distinctions lie in synaptic weight fluctuations, a trend observed in all comparisons between RSTDP and STDP simulations. Additionally, a subtle difference is observed in the postsynaptic minimum value, which reaches -0.001 in these instances.

RSTDP reaches lower values than STDP primarily due to its integration of reward signaling into the synaptic weight adjustment mechanism. This reward-modulated learning introduces an additional layer of complexity, where synaptic changes are not solely based on the timing of pre- and postsynaptic spikes but also on the expectation and receipt of rewards. Consequently RSTDP may fine-tune synaptic adjustments more conservatively to align with rewarded behaviors, potentially resulting in overall lower synaptic weights compared to STDP, which adjusts weights based purely on spike timing correlation. Additionally, with the inclusion of reward signals, RSTDP might promote a more selective strengthening or weakening of synapses, leading to a reduced overall synaptic efficacy when compared to the broader adjustments seen with STDP.

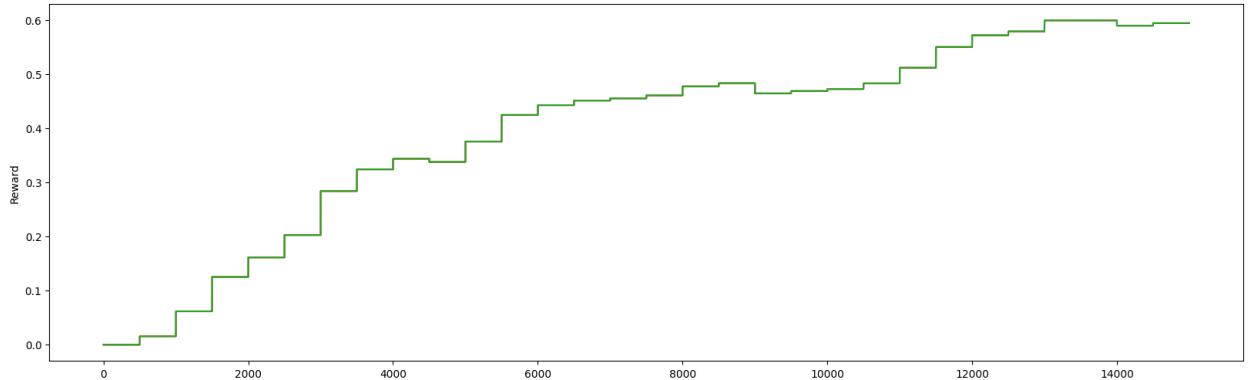


Figure 34 Synaptic reward during training time course with RSTDP learning and inactive neurons.

Another observed fact is the consistent presence of both post- and presynaptic traces despite reward-modulated learning, this reflects the underlying neural mechanisms that remain active regardless of the learning paradigm. The traces indicate ongoing neural activities; it represents the action potentials or neural firing from the sending neurons, an essential component of neural communication and synaptic plasticity. It also illustrates the receiving neurons' responses, including hyperpolarization or depolarization events, key for synaptic strength adjustment and neuronal network modulation.

Even with the introduction of reward signals in RSTDP, the fundamental process of synaptic transmission and its basic representational method (via traces) do not change. The presynaptic and postsynaptic traces, thus, signify continuous electrical activities essential for both STDP and RSTDP-driven synaptic modifications, highlighting the unaltered core aspects of neural interaction despite the addition of reward-based learning complexities.

As expected, the synapse connecting the two inactive neurons remains unchanged throughout. Its weight retains its initial value. The presynaptic trace consistently shows a value of zero during training, attributable to the zero firing rate of the input neuron. Similarly, the postsynaptic trace is also zero, reflecting the inactivity of the output neuron (**Figure 33**).

In RSTDP, the reward trend reaches a value of 1 when all neurons are active. However, when two inactive neurons are added to the input and output layers in RSTDP, the reward trend follows an upward trend towards 0.6 (**Figure 34**).

The observed reward trend dynamics highlight the intricate balance between neuronal activity and learning outcomes modulated by rewards. With all neurons active, the network is fully utilized, meaning every neuron contributes to processing the information and achieving the task. This optimal use aligns with the core objective of RSTDP, which is to enhance learning and memory by adjusting synaptic strengths in a way that maximizes reward. When the network operates at full capacity, and every component acts in concert towards the goal, the likelihood of achieving the task correctly and efficiently is maximized, thus earning the highest possible reward (value of 1).

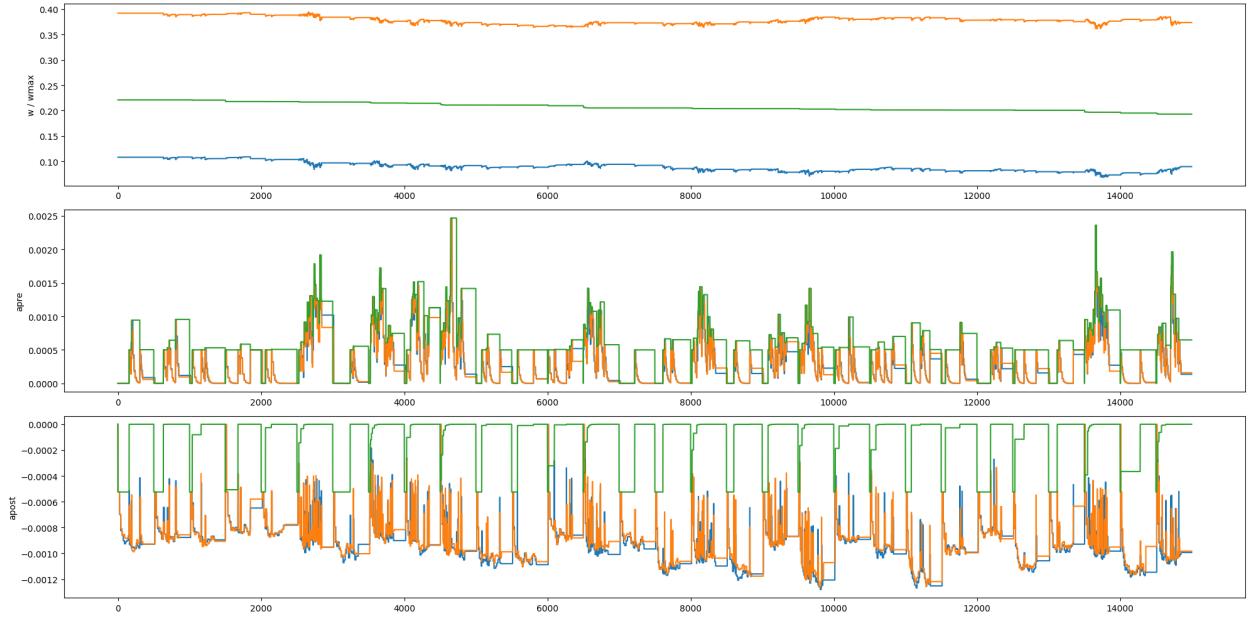


Figure 35 Synapse dynamics. The top panel shows the output neurons weights over time. The middle panel displays the presynaptic trace and the bottom panel displays the postsynaptic trace. All neurons have minimum firing rate and trained with RSTDP learning method.

Introducing inactive neurons into the input and output layers alters the network's ability to process information and achieve tasks effectively. In RSTDP, rewards are used to modulate the synaptic plasticity based on the temporal difference in spikes between presynaptic and postsynaptic neurons, with the goal of reinforcing pathways that lead to successful task completion. Inactive neurons disrupt these pathways, leading to a less efficient information flow and, consequently, a lower reward. The reason for the upward trend to 0.6, rather than a static lower value, suggests that while the network's performance is impacted, RSTDP algorithms can still somewhat adapt to these suboptimal conditions by finding alternative pathways or strategies, albeit not as successfully as when all neurons are engaged.

The upward trend to 0.6 in the presence of inactive neurons also reflects the adaptive aspect of RSTDP. This learning mechanism is designed not only to strengthen the most efficient neural pathways but also to explore and potentially reinforce alternative pathways when the optimal ones are compromised (e.g., by inactive neurons). This adaptability ensures that learning and task performance can still occur under less-than-ideal conditions, albeit at a reduced reward level, as the system tries to compensate for the inefficiencies introduced by the inactive neurons.

Task 7: Set minimum firing rate (RSTDP)

As observed in **Figure 35**, the synaptic weight of the third neuron also varies throughout the simulation period, with the postsynaptic trace fluctuating due to the implementation of the minimum firing rate setting. Furthermore, in **Figure 36**, activity is reignited between two previously inactive neurons, demonstrating that the method effectively ensures that no neurons

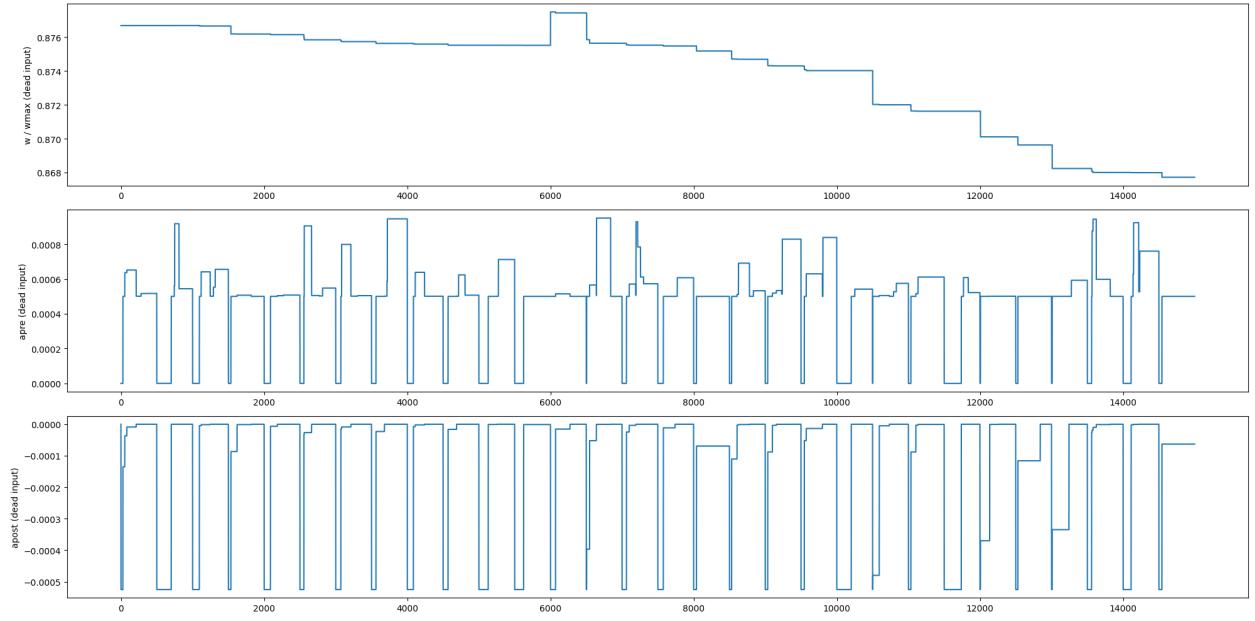


Figure 37 Synapse connecting the two previously inactive neurons in input and output layers learned with RSTDTP.

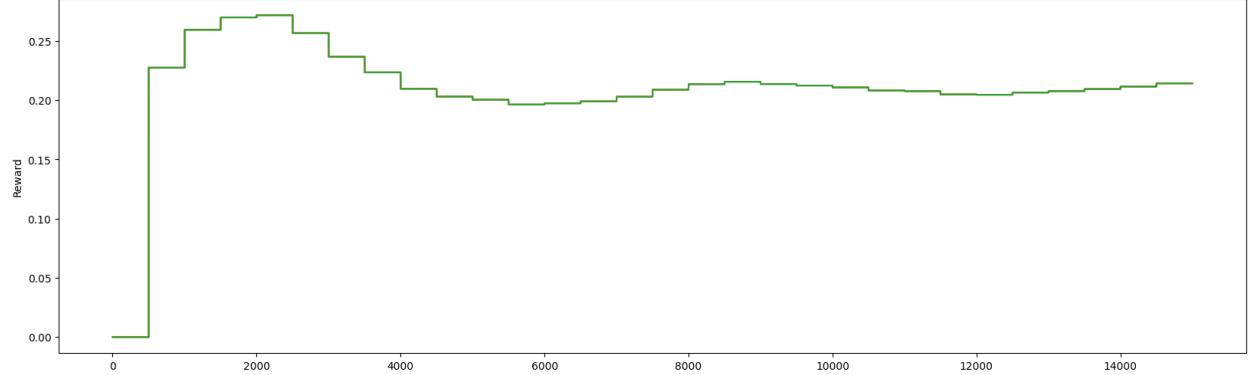


Figure 36 Synaptic reward during training time course with RSTDTP learning and minimum rate applied.

remain inactive. This shows the efficiency of the approach in maintaining neural network activity and preventing inactivity within the network.

Activating the mechanism appears to have a lesser impact on the overall synaptic weights compared to the STDP model. Furthermore, a comparison between **Figures 25** and **36** indicates that the change in synaptic weights is more significant in the STDP model, ranging from 0.81 to 0.87, whereas in the RSTDTP model, this range is narrower, from 0.86 to 0.87.

Setting a minimum firing rate for all neurons contributes to its overall activity. Imposing a minimum firing rate ensures that all neurons are engaged in the network's task processing. This increased participation can enhance the overall efficiency of information processing and decision-making pathways. As a result, the network should theoretically operate more cohesively, with a higher likelihood of achieving task objectives more efficiently and effectively. This is seen in **Figure 37**. Note that the scale for the minimum and maximum range in this figure is adjusted differently compared to previous simulations, where a value of 0.25 now corresponds to what was considered as 1 in all earlier scenarios.

While the reintroduction of all neurons into active processing does not guarantee a return to the perfect reward value of 1 (primarily because the preset minimum firing rate might not perfectly match the network's optimal operational dynamics needed for maximal reward), it does indicate a movement towards higher reward values. The exact value would depend on how well the introduced minimum firing rate aligns with the optimal activity level for each task and how efficiently the network can adapt its synapses under RSTDP to the new operational paradigm.

In essence, by setting a minimum firing rate for all neurons, you can anticipate an increase in the reward trend from its previous state, signifying an improvement but possibly not reaching the idealized reward scenario of complete optimization. This adjustment promotes a more active and engaged network, which RSTDP can utilize to enhance learning outcomes through better-informed synaptic modifications.

Task 6: Parameters effect

Poisson distribution rate in STDP learning

Increasing the maximum rate of the Poisson distribution leads to a higher frequency of input spikes. For STDP, which depends on the precise timing between pre- and post-synaptic spikes, a higher input rate can intensify the weight update mechanism. This is because there are more opportunities for the temporal correlations between input and output spikes to be exploited for synaptic strengthening or weakening.

The effect on learning can be dual. On one hand, it might accelerate learning by increasing the number of potentiation or depression events. On the other hand, it could also lead to more noise in the learning signal, potentially destabilizing synaptic weights if not appropriately regulated. The optimal rate for this simulation has been set to 63Hz for the pixel with 255. Intensity.

Poisson distribution rate in RSTDP learning

RSTDP adjusts the weight change mechanism of STDP by incorporating a reward signal. An increase in the Poisson max rate could lead to more pronounced weight adjustments in response to rewarded behaviors, given the more frequent spike events.

The RSTDP model is designed to enhance the model's sensitivity to rewarded outcomes. With higher input rates, this mechanism might effectively modulate synaptic plasticity towards more optimal behaviors or patterns that are associated with rewards. However, similar to STDP, an excess of input spikes could introduce noise, potentially requiring careful tuning of the learning parameters to maintain stable and meaningful learning.

Max synapse weight in STDP learning

A higher max weight allows for a broader range of synaptic strength adjustments. This can lead to stronger connections between neurons that consistently exhibit correlated firing patterns,

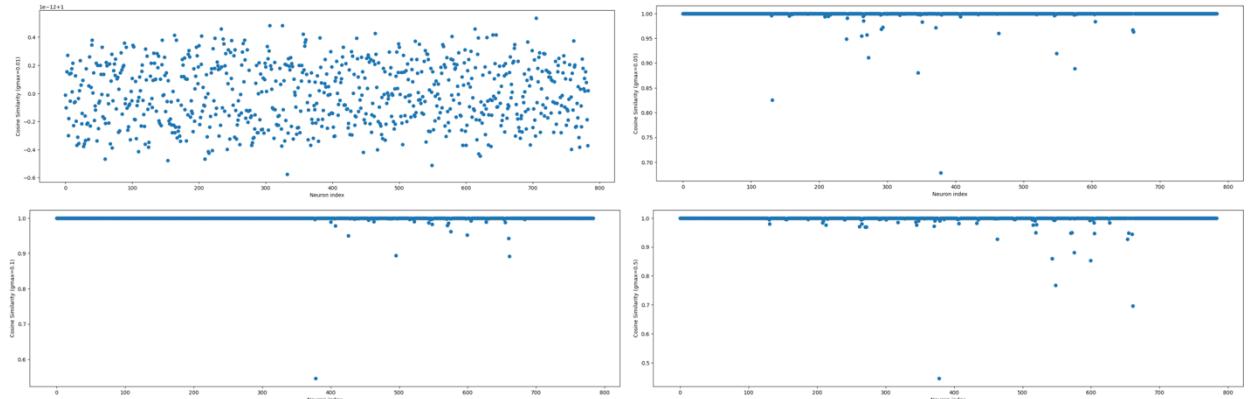


Figure 39 Cosine similarity across four different maximum weight values (0.01, 0.05, 0.1 and 0.5) with STDP.

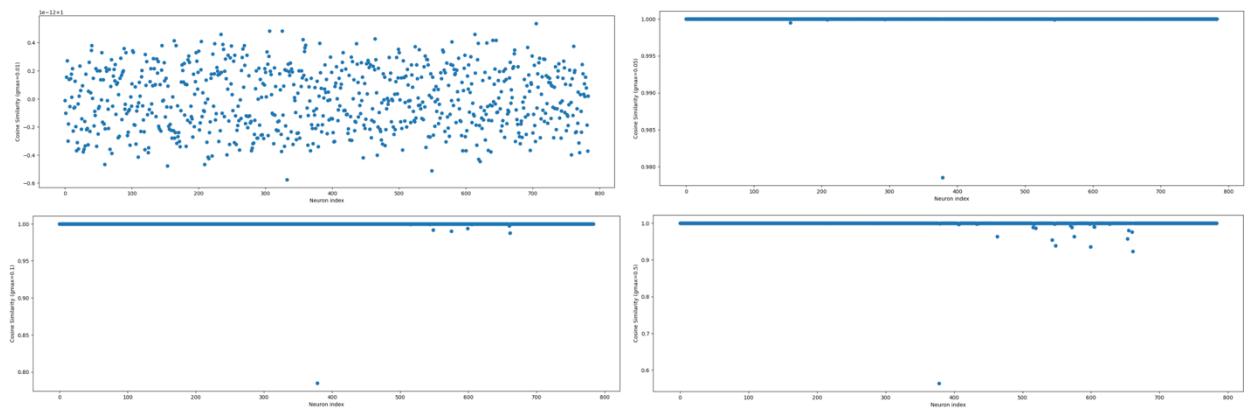


Figure 38 Cosine similarity across four different maximum weight values (0.01, 0.05, 0.1 and 0.5) with RSTDP.

potentially enhancing the network's ability to learn and distinguish between different inputs, such as MNIST digits.

While increased weight limits provide neurons with a greater capacity to strengthen their connections, there's also a risk of synaptic weights saturating too quickly. This could diminish the network's sensitivity to further learning since additional significant patterns in the input data might not cause further synaptic modifications once the weights have reached their maximum. **Figure 39** displays the cosine similarity across four different maximum weight values, each represented on the x-axis.

Max synapse weight in RSTDP learning

In the RSTDP framework, rewards modulate the amount of synaptic change allowed by the STDP rule. A higher maximum weight could amplify the effects of rewards, enabling more significant synaptic changes in response to rewarded behaviors. This could facilitate faster learning of behaviors associated with high rewards.

However, significantly increasing the maximum synaptic strength might lead to overfitting, especially if the reward signals are too strong or too frequent. The network might overemphasize the learning of specific patterns associated with rewards at the expense of generalization. **Figure**

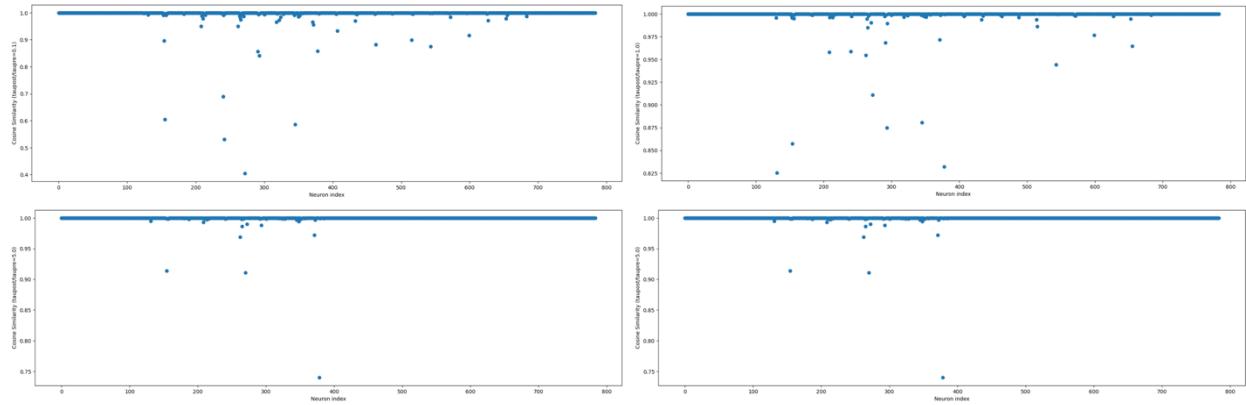


Figure 40 Cosine similarity across four different $\tau_{\text{post}}/\tau_{\text{pre}}$ ratio (0.1, 1, 5 and 10) with STDP.

38 displays the cosine similarity across four different maximum weight values, each represented on the x-axis.

Input format in STDP/RSTDP

Earlier in the text, we illustrated how the difference image is presented to the input in two distinct manners: first, as a whole, and then in segments that gradually overlap more during the simulation. This variation in presentation and its effects on STDP and RSTDP are analyzed and depicted across various figures. Please refer to the respective sections for a detailed examination of these dynamics.

taupost/taupre ratio in STDP

These two variables represent the time constants for the decay of post-synaptic and pre-synaptic traces, respectively. These traces contribute to determining the amount of synaptic modification (either potentiation or depression) based on the relative timing of pre- and post-synaptic spikes.

Initially, increasing the *taupost/taupre* ratio can adjust the balance between long-term potentiation (LTP) and long-term depression (LTD), potentially making the network more sensitive to the specific temporal patterns of spikes. An increased *taupost* makes the network more likely to strengthen synaptic weights for post-synaptic spikes following pre-synaptic spikes over longer time intervals, enhancing its ability to capture causal relationships over longer durations.

The modification of these parameters directly impacts the learning dynamics. A network might become more adept at learning from temporal patterns spanning wider time windows, potentially improving its performance on tasks that require integration over time or the detection of long-range temporal correlations.

While initial adjustments to the ratio can significantly impact learning, beyond a certain threshold, further increases do not continue to change the overall cosine similarity between the learned patterns and target patterns (**Figure 40**). This plateau can be attributed to several factors:

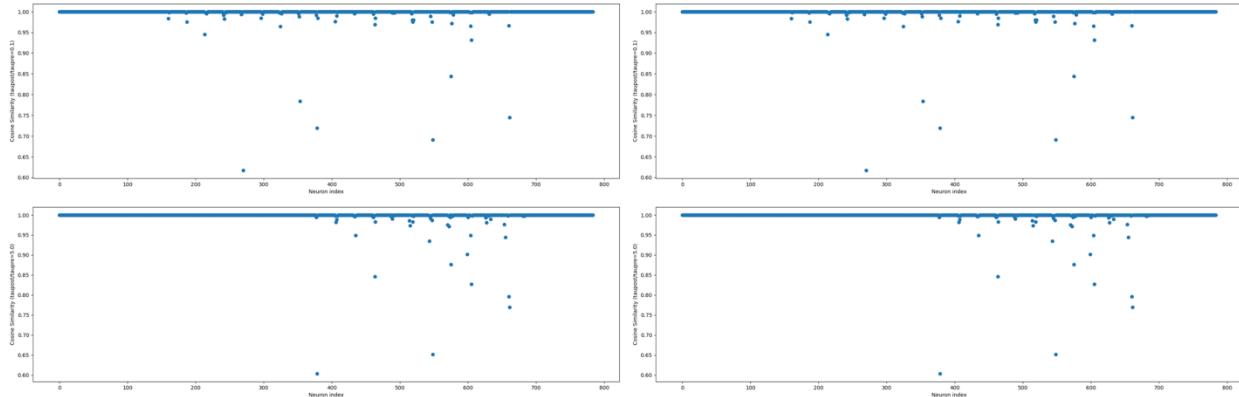


Figure 41 42 Cosine similarity across four different $\tau_{\text{post}}/\tau_{\text{pre}}$ ratio (0.1, 1, 5 and 10) with RSTDp.

1. *Saturation of Learning Window*: Beyond a certain point, the extended time constants may exceed the typical range of relevant inter-spike intervals. In practical terms, this means the additional window expansion doesn't capture more meaningful temporal relationships because those relationships fall outside the typical data-pattern time scale.
2. *Noise Introduction*: Extending the decay times too much might also introduce noise into the learning process, as synaptic changes could then be influenced by spikes with less precise temporal correlation. This could muddy the synaptic adjustments, leading to a plateau or even decrease in the quality of learned representations.
3. *Optimal Time Constants*: There exists an optimal range for τ_{post} and τ_{pre} within which the neural network can effectively capture the dynamics of the input spike patterns. Beyond this range, the neural network's ability to discern useful temporal patterns does not significantly improve because the additional information captured does not contribute to or may even detract from the structured patterns it has learned.

taupost/taupre ratio in RSTDp

Understanding the effect of altering the $\tau_{\text{post}}/\tau_{\text{pre}}$ ratio in RSTDp, and why its impact on learning outcomes, such as overall cosine similarity, might plateau or be less sensitive compared to standard STDP, requires a detailed examination.

Increasing the $\tau_{\text{post}}/\tau_{\text{pre}}$ ratio in RSTDp affects the network's ability to encode and process temporal information. Initially, this can enhance learning by:

A higher ratio extends the time window for linking post-synaptic activity to prior pre-synaptic spikes, potentially strengthening the encoding of temporal sequences and improving the association between neural activity and rewards.

Lengthening τ_{post} makes it easier to temporally associate synaptic changes with delayed rewards, leveraging the "eligibility trace" concept where synaptic candidates for modification are tagged for later reward-based adjustment.

The phenomenon where further increases in the $\tau_{\text{apost}}/\tau_{\text{apre}}$ ratio cease to improve or even perturb learning outcomes, such as cosine similarity between learned and target patterns, can be attributed to the fact that there is an optimal temporal encoding window for the association of inputs to rewards. Beyond this, extending the window may capture irrelevant temporal information, diluting the specificity of the associations and potentially impeding the learning process.

In addition, in RSTDP, the efficacy of learning is subject to the dynamics of reward delivery. If synaptic changes are too broadly or weakly associated with rewards, due to an overly extended temporal window, the precision of learning diminishes. The network becomes less able to discern the specific events leading to rewards, thereby stalling improvements in learning outcomes.

Contrary to initial expectations, RSTDP may exhibit less sensitivity to changes in the $\tau_{\text{apost}}/\tau_{\text{apre}}$ ratio than STDP due to its integrated reward mechanism, which can be seen through:

1. Reward Modulation Buffering: The reward signals in RSTDP serve to modulate the strength and direction of synaptic changes, potentially buffering the network against the impacts of temporal parameter adjustments. The reward-based modulation might mean that only synaptic changes aligned with positive reward outcomes are reinforced, limiting the impact of further extending the temporal window.
2. Complex Temporal Dynamics: RSTDP entails a more complex interplay of temporal dynamics and reward processing compared to STDP. This complexity could mean that the simple extension of $\tau_{\text{apost}}/\tau_{\text{apre}}$ ratio has a diluted effect, as the network's learning is shaped by a combination of spike timing and the timing and magnitude of received rewards. The consequence is a dampened sensitivity to parameter changes, provided that the reward signals effectively guide synaptic modifications towards desired outcomes.