Computational Neuroscience

Project No. 4, Report

Name: Mahroo Hajimehdi Student No: 610399198

Introduction

The primary goals of this project include gaining an in-depth understanding of the mechanisms that exist between neurons within a single layer and examining the impact of these mechanisms on the learning process, specifically STDP learning scheme. By investigating these neural interactions, we aim to uncover how neurons communicate and influence each other's activities, thereby elucidating their collective role in learning dynamics.

For implementation, several libraries have been utilized, including Pytorch, PymoNNtorch and CoNeX. The network structure comprises a spiking neural network (SNN) with two layers: a fully connected input layer featuring 10 neurons and an output layer containing 2 neurons. Each neuron follows Leaky Integrate-and-Fire (LIF) dynamics and the input is simply a tensor consisting of ten binary elements (0s and 1s), which are presented to the input layer across different learning epochs. In addition, The learning scheme employed throughout the report is Spike-Timing-Dependent Plasticity (STDP).

Task 1: STDP Learning for 1D Input

First, we need to generate two distinct binary patterns to feed into the network during different learning epochs. The LIF model for the first layer is slightly modified; its activity is directly influenced by the spiking pattern of the input tensor rather than by changes in current or voltage. To facilitate further tasks, we have added `SpikeTrace()` and `NeuronAxon()` classes to the neuron groups.

The synaptic weights are initially set using a normal distribution with a mean of 0.2 and a standard deviation of 0.1. The STDP rule employs soft bounds for positive and negative limits, ensuring that throughout the learning process, the weights remain within the interval of 0.0 to 1.0.

Results

The results of the learning process are presented using raster plots of the two layers, with two arbitrary input patterns, as shown in **Figure 1**. The synaptic weight changes can also be seen in **Figure 2**. No other significant modifications within the layer are applied at this step, we are simply establishing a foundation for the tasks ahead.

If the weights were not correctly modified by the network's learning process, we would expect random bursts of spikes in the output whenever the input is present. However, we can observe that neuron 10 (the first neuron in the output layer) exhibits a spike pattern closely related to the first

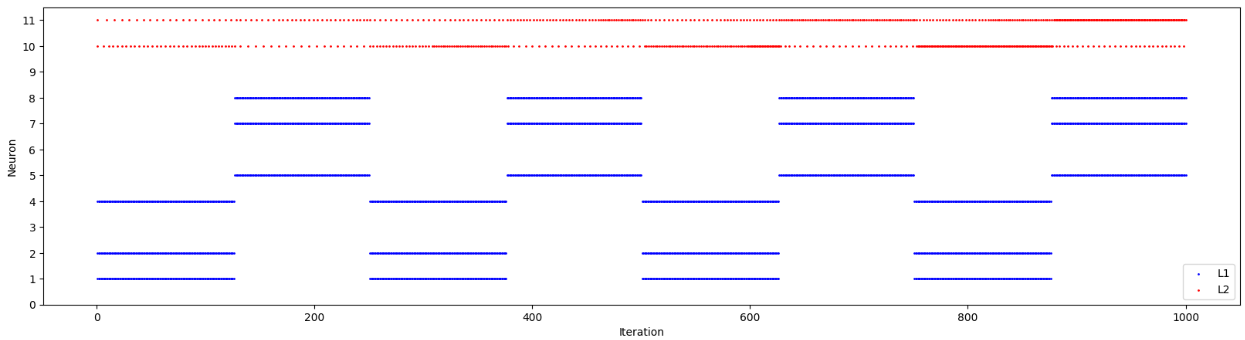

Figure 1 Network activity on distinct pattern.

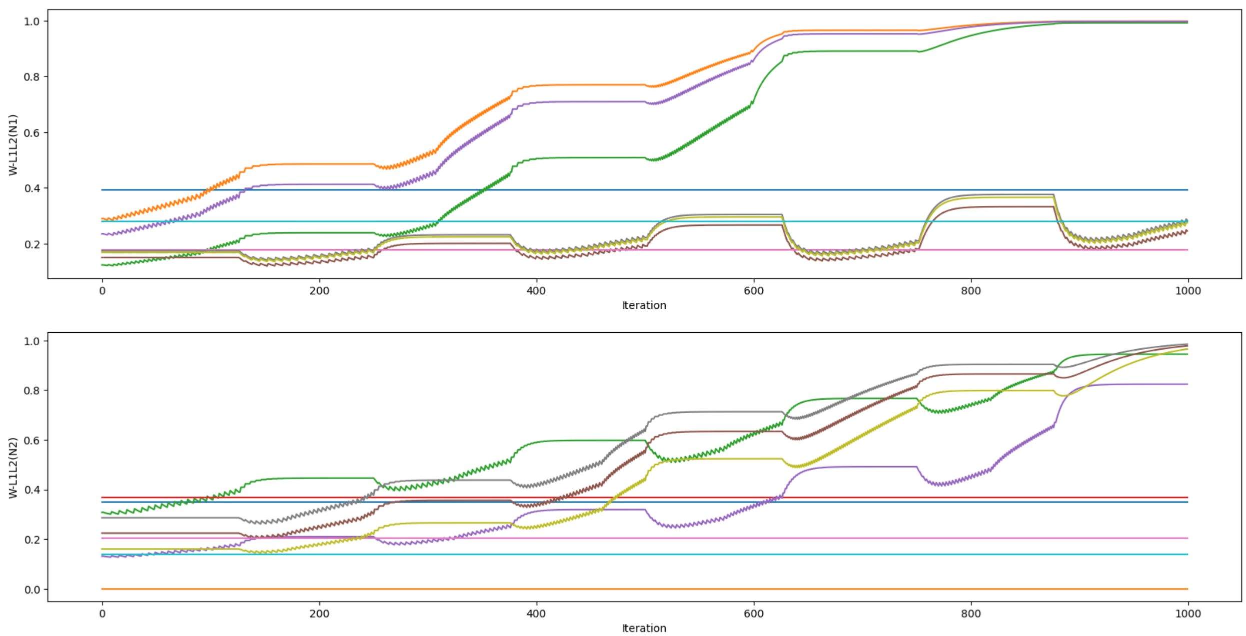

Figure 2 Weight changes for network of 10 input neurons and 2 output neurons. top panel: weight changes of synapses between layer 1 and output neuron 1, bottom panel: weight changes of synapses between layer 1 and output neuron 2.

input pattern. Although it shows some activity when the second pattern is presented, it can be primarily associated with the first input pattern. Similarly, neuron 11 (the second neuron in the output layer) is trying to learn the second input pattern. When the first input pattern is presented, the activity of this neuron drops slightly compared to the second pattern.

Task 2: Adding Lateral Inhibition to the 2nd Layer

a) Distinct patterns

Lateral inhibition is a neural mechanism that enhances contrast and sharpens signal boundaries in neural processing. In biological neural systems, it refers to the ability of an excited neuron to reduce the activity of its neighbors. This mechanism ensures that the most strongly activated neurons suppress the activity of their neighboring neurons, leading to a clearer and more precise signal. It’s usage includes the following:

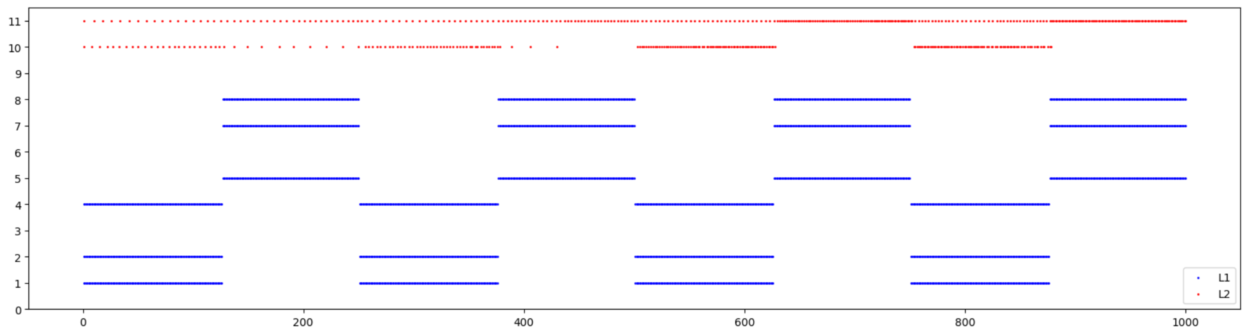

Figure 3 Network activity on distinct pattern with Lateral Inhibition mechanism added to layer 2.

1. **Noise Reduction**: By suppressing minor signals from neighboring neurons, lateral inhibition helps in reducing noise and improving the clarity of the significant signals.
2. **Efficient Coding**: Lateral inhibition encourages a sparse representation of information. Only the most relevant and strong signals are propagated, which leads to more efficient information processing and storage.
3. **Signal Decorrelation**: By minimizing redundancy among the signals of neighboring neurons, lateral inhibition aids in creating a more distinct and unique coding scheme. This is particularly useful in pattern recognition tasks which is our case of usage
4. **Enhanced Differentiation**: It allows spiking neural networks to discriminate between similar patterns by sharpening the boundaries of active regions in the network.
5. **Temporal Precision**: Lateral inhibition improves the temporal precision of spiking neurons, essential for tasks that require the detection of precise timing differences, crucial in auditory and sensory processing.

We define a synapse group between the neurons of the second layer to apply this strategy. The function used is `MyLateralDendriticInput(),` which has minor changes compared to the CoNeX's `LateralDendriticInput()` behavior. The results of the simulation are shown in **Figure 3** and **Figure 4**.

Results

As expected, neuron 10 exhibits a spike pattern strongly correlated with the first input pattern. Initially, it shows some activity when the second pattern is presented, but in the late stages of training, almost no activity is observed when the second pattern is introduced to the network (**Figure 3**). This effect is due to the lateral inhibition process, where only strong activities are propagated to the subsequent stages of training.

Neuron 11, on the other hand, is learning the second input pattern. When both input patterns are presented, the activity of this neuron is weak in the early stages of training. However, as training progresses, the activity increases whenever the second pattern is present. If the training process goes beyond the shown 1000 iterations, we can expect minimal or no activity for the first pattern.

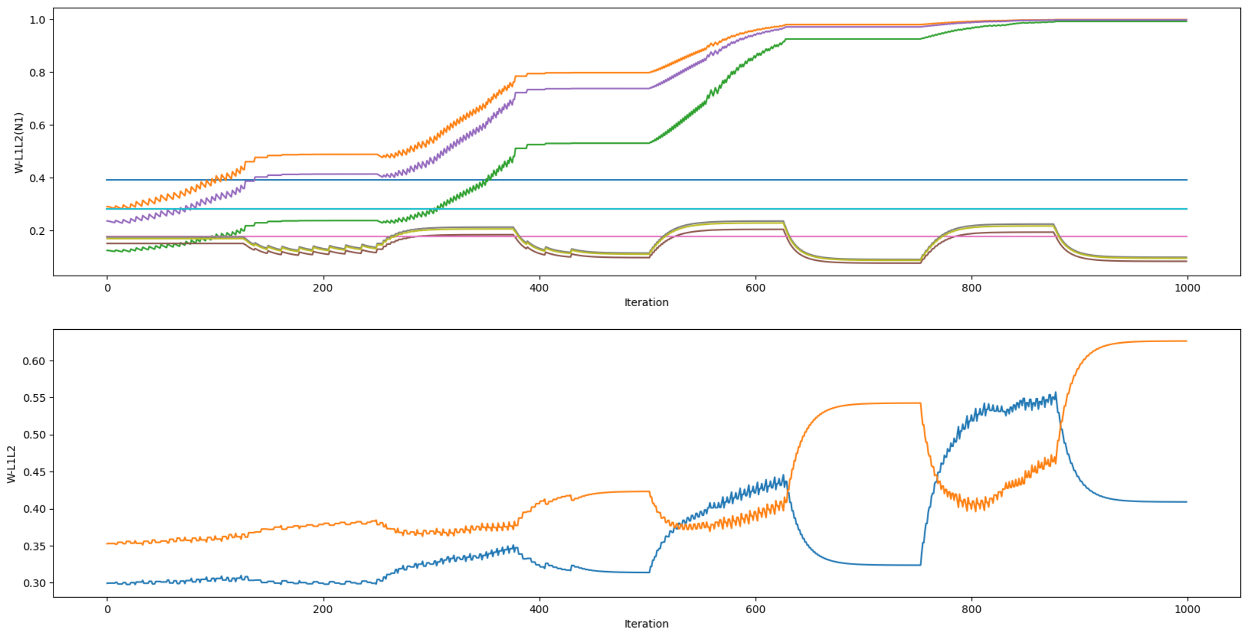

Figure 4 Weight changes for network with Lateral Inhibition mechanism added to layer 2. top panel: weight changes of synapses between layer 1 and output neuron 1, bottom panel: weight changes of synapses between layer 1 and output neuron 2

The improved results from the previous simulations are directly related to the lateral inhibition process. The two neurons inhibit each other's activity if it's too weak, leading to better discrimination of distinct patterns. This can be clearly seen in the second panel of **Figure 4**.

b) Similar patterns

The network successfully distinguished between two completely different patterns. We will now generate patterns with similar parts to test whether the network can still recognize them.

Results

As shown in **Figures 5, 6, 7, and 8**, when the patterns become more similar, the network's ability to effectively distinguish between them diminishes. Throughout the simulation, neuron 11 tends to respond primarily to the similar parts of the patterns, leaving the recognition of the unique characteristics of one pattern to the other neuron. As the similarity between the patterns increases, one neuron initially attempts to identify the input pattern. However, as the simulation progresses, neuron 10 also struggles to capture the distinct patterns, resulting in both output neurons focusing on the similar parts during the entire training process.

As for why this behavior is observed, we can give several explanations:

When the two input patterns are distinctly different, each output neuron can reliably distinguish between the two patterns. Each distinct input pattern likely activates a unique subset of input neurons. This unique combination of activated input neurons leads to unique spiking patterns that the output neurons can learn to recognize and differentiate. In addition, The network, through STDP, adjusts the synaptic weights such that each output neuron becomes highly responsive to one specific pattern and not the other. This helps in clear pattern recognition and separation.

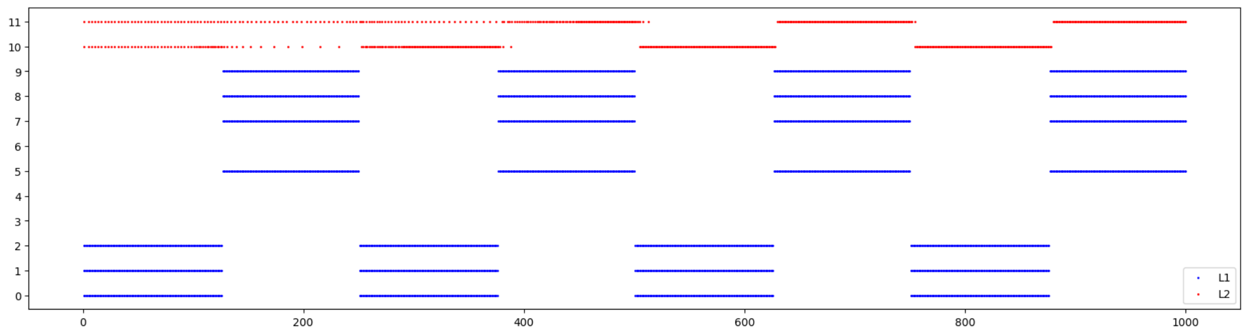

Figure 5 Network activity on similar pattern with Lateral Inhibition: zero similarity.

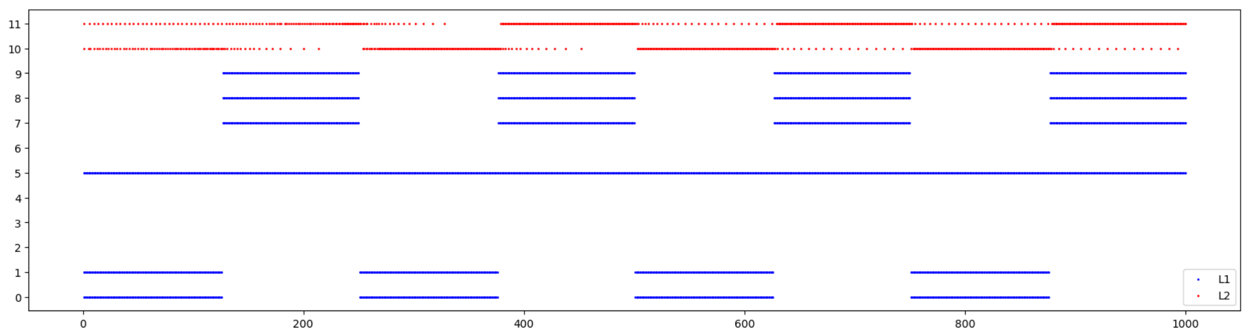

Figure 6 Network activity on similar pattern with Lateral Inhibition: similar activity on neuron 5.

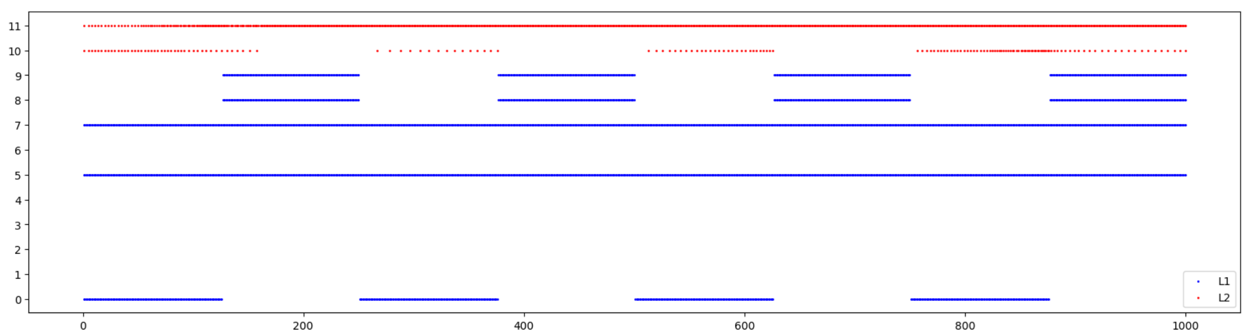

Figure 7 Network activity on similar pattern with Lateral Inhibition: similar activity on neuron 5 and 7.

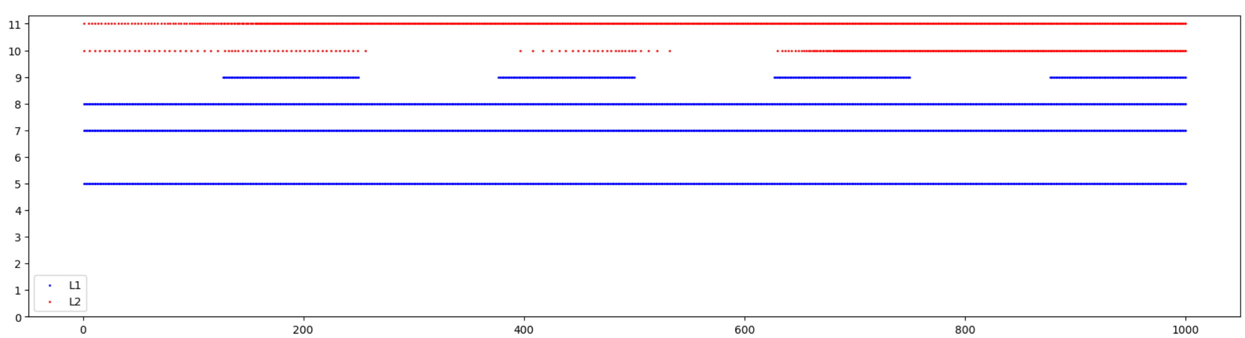

Figure 8 Network activity on similar pattern with Lateral Inhibition: similar activity on neuron 5, 7 and 8.

As the similarity between the two input patterns increases, they start to activate more overlapping sets of input neurons. This overlap reduces the distinctiveness of each pattern at the level of the input neurons. The overlapping activations lead to more similar overall spike patterns reaching the output neurons, making it harder for the network to differentiate between the two patterns. So this slight increase in similarity may lead to the situation where one of the output neurons becomes more dominant in its response. This happens because the synaptic strengths might be such that one neuron consistently receives slightly stronger input than the other for both patterns, leading it to be the more active neuron by default.

As a result, The output neurons lose their selectivity because the patterns do not provide enough distinguishing features to maintain differentiated responses. The network's synaptic weights might not be tuned to handle such high similarity, causing both neurons to activate for both patterns.

But we do not expect both neurons to be active at the same time because of the lateral inhibition mechanism, so the observation suggest that the lateral inhibition mechanism is not strong enough for neurons to suppress each other's activity adequately. To overcome this issue we increased the strength of this mechanism for the most similar patterns and the result is shown in **Figure 9**.

A Suggestion

We suggest that using Adaptive Exponential Leaky Integrate-and-Fire (Adaptive ELIF) instead of standard LIF neurons could mitigate this challenge, as similar patterns cause one or more neurons to be consistently active. This adaptation could shift the neurons' focus from the identical parts of the patterns to the distinguishing features. In addition, the exponential term in the AELIF model makes the neuron's response to inputs more nonlinear. This nonlinearity can help in differentiating between similar patterns by producing distinct membrane potential trajectories for each pattern. However, this is a hypothesis that requires further testing.

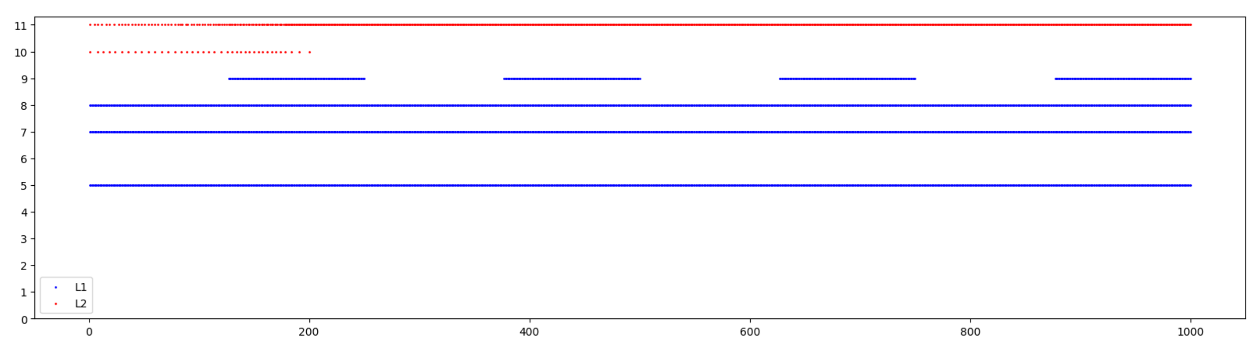

Figure 9 Network activity on similar pattern with Lateral Inhibition: similar activity on neuron 5, 7 and 8. The synaptic strength is increased to prevent the simultaneous activity of 2 output neurons.

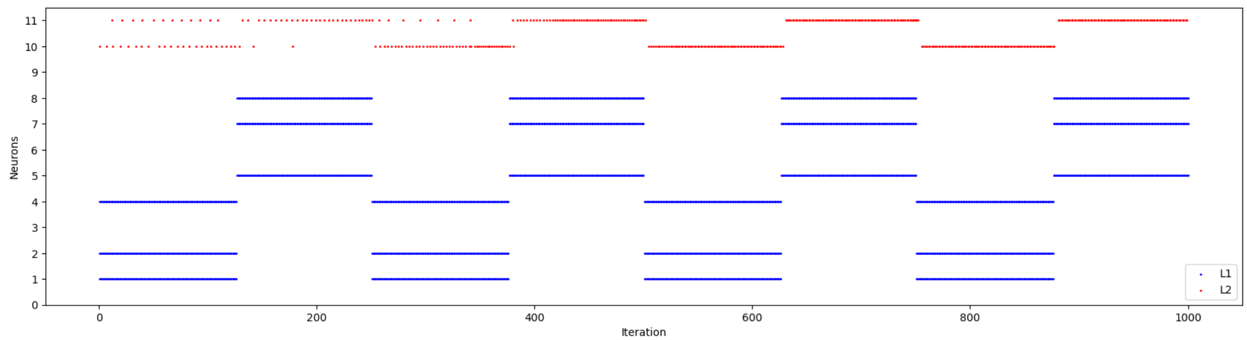

Figure 10 Network activity on distinct pattern with K-WTA mechanism added to layer 2.

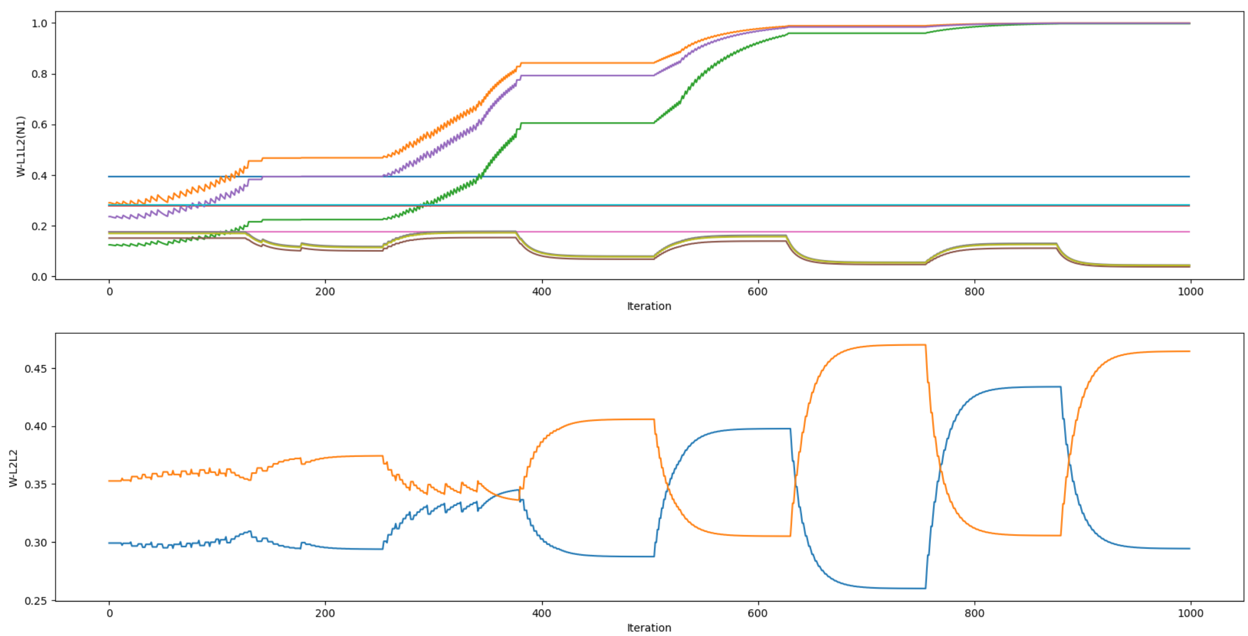

Figure 11 Weight changes for network with K-WTA mechanism added to layer 2. top panel: weight changes of synapses between layer 1 and output neuron 1, bottom panel: weight changes of synapses between layer 1 and output neuron 2

Task 3: Adding K-WTA to the 2nd Layer

The *K-Winner-Takes-All (K-WTA)* mechanism is a neural competition strategy commonly used in SNNs and other types of neural networks. It functions by allowing only the top K neurons with the highest activation levels to remain active while inhibiting the rest. Inhibitory signals are employed to suppress the neurons that do not belong to the top K active set.

K-WTA promotes sparse representations by ensuring that only a few neurons are active at any time. This is beneficial for reducing redundancy and overlap in neural representations. In other words, Ensuring that only the most relevant neurons fire enhances the network's ability to recognize and differentiate between similar patterns. As a result, By suppressing less significant activations, K-WTA ensures that only the strongest signals are considered, thereby reducing the impact of noise.

We added K-Winner-Takes-All (K-WTA) mechanism with K=1 into our SNN, which already employed STDP and lateral inhibition, so it is expected to yield several benefits and changes in network dynamics. The result is shown in **Figure 10** and **Figure 11**.

Results

As **Figure 10** suggests, we can see a clear discrimination of patterns happening earlier in the training process compared to Task 2. Comparing **Figure 11** and **Figure 4** we can observe differences in the synaptic weight dynamics for two output neurons in Task 2 and Task 3:

1. **Fluctuation Reduction:** Weight fluctuations are smaller when K-WTA is applied.
2. **Greater Weight Differences:** The difference between the weights of the two output neurons is more pronounced when K-WTA is used.
3. **Weight Intervals:**

    Without K-WTA: Weights are between 0.3 and 0.6.

    With K-WTA: Weights are between 0.25 and 0.45.

The **first** observation can be explained by the fact that STDP modifies the synaptic weights based on the relative timing of spikes. When only the most significant neurons (determined by K-WTA) are considered, the synaptic adjustments are more consistent and less prone to the variability that might come from fluctuating sub-dominant neurons. Therefore, the weights exhibit less fluctuation.

The **second** observation is because the combination of lateral inhibition and K-WTA strengthens the inhibitory effects. This results in a more pronounced differentiation between the neurons because the K-WTA ensures that only the most relevant neurons are active, leading to a clearer separation in their functional roles and consequently greater weight differences.

As for the **third** observation, In the absence of K-WTA, the weight updates are influenced by a wider range of neuronal activations. The lateral inhibition promotes contrast but does not restrict the number of active neurons to the same extent. Therefore, the weights can reach higher values as more neurons contribute to the weight update process.

On the other hand, K-WTA restricts the weight updates to only the top K neurons. This concentrated learning effort means that although the learning is more stable, it is also more competitive. The limited set of top neurons means there's less chance for extreme weight excursions because only the most prominent spikes contribute to STDP. This prevents weights from becoming too large (as seen in the case without K-WTA), maintaining them in a more moderated range.

In conclusion, with K-WTA in place, STDP can more effectively fine-tune the synaptic weights. Only the connections corresponding to the most significant neurons (as determined by K-WTA) will be strengthened or weakened, leading to more efficient learning. Neurons that consistently fall outside the top KK may receive less synaptic strengthening over time, leading to a natural pruning process that aligns with the network's focus on salient features.

When combined with lateral inhibition, K-WTA provides an additional layer of suppression. Lateral inhibition works by locally suppressing neighboring neurons, while K-WTA globally ensures only the top activations remain, leading to a more robust inhibition mechanism. The dual inhibitory strategies can create a balanced network where both strong contrast (via lateral inhibition) and focused activation (via K-WTA) coexist, enhancing overall network performance.

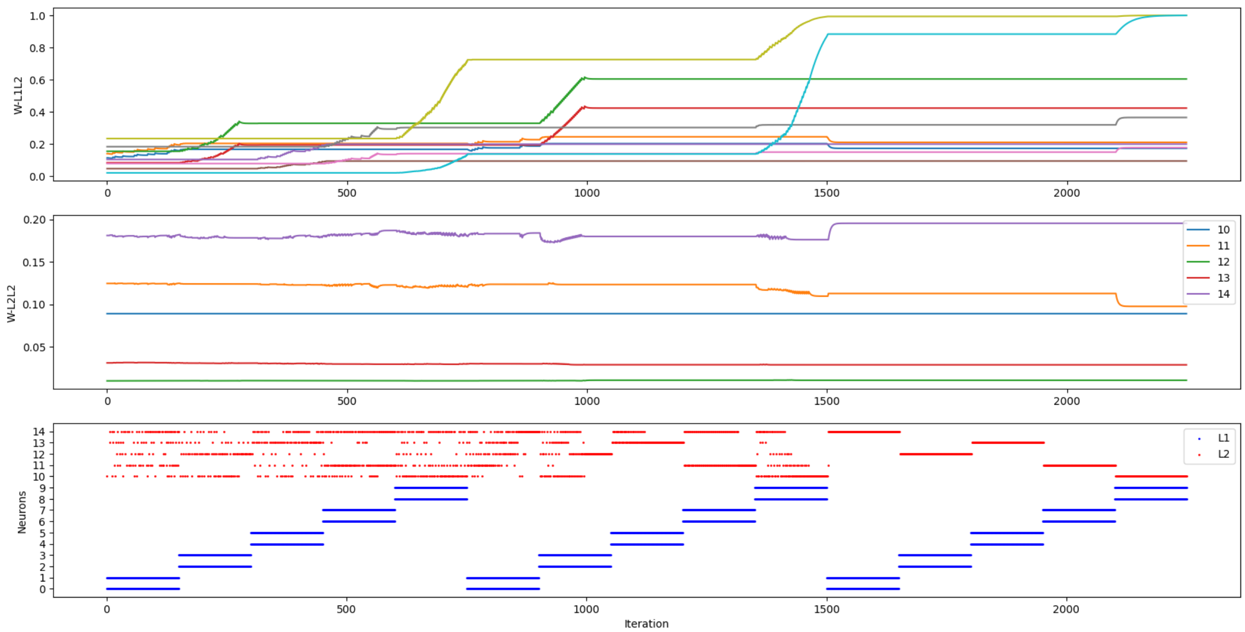

Figure 12 Network activity on distinct pattern with VBH mechanism added to layer 2 synapses. top panel: weight changes of excitatory synapses between input and output neurons, mid panel: weight changes of inhibitory synapses between output neuron. bottom panel: raster plot of 10 input and 5 output neurons activity with 5 distinct pattern input.

Task 4: Adding Homeostasis to the 2nd Layer

Homeostasis in neural networks refers to mechanisms that maintain stability and balance in the network's activity. In the context of SNNs, homeostasis ensures that neurons do not become overly active or inactive, which is crucial for maintaining efficient and robust information processing. One specific form of homeostasis is **Voltage-Based Homeostasis**, which regulates the activity of neurons based on their membrane potential.

The membrane potential is the difference in electric potential between the inside and outside of a neuron. It determines the neuron's readiness to fire a spike. *Voltage-based homeostasis* monitors and adjusts this potential to keep neuronal activity within a desired range. This mechanism ensures that neurons neither become too excitable nor too inhibited, promoting stable network dynamics.

1. Training with 5 Output Neurons and 5 Distinct Patterns

In this step we generate 5 different pattern and increase the number of output neurons from 2 to 5 and train the network. Voltage-Based homeostasis scheme is used to balance the network and improve its performance. The result is shown in **Figure 12**.

Results

After 1500 iterations the 5 output neurons can each distinguish between the 5 patterns (**Figure 12**). Although the first pattern seems to be recognized by neuron 14 too, it is mainly correlated with neuron 12. We designed the homeostasis framework to maintain all neurons voltage falls in the interval (-60, -40) which is 10 mV above and below the LIF neurons' threshold.

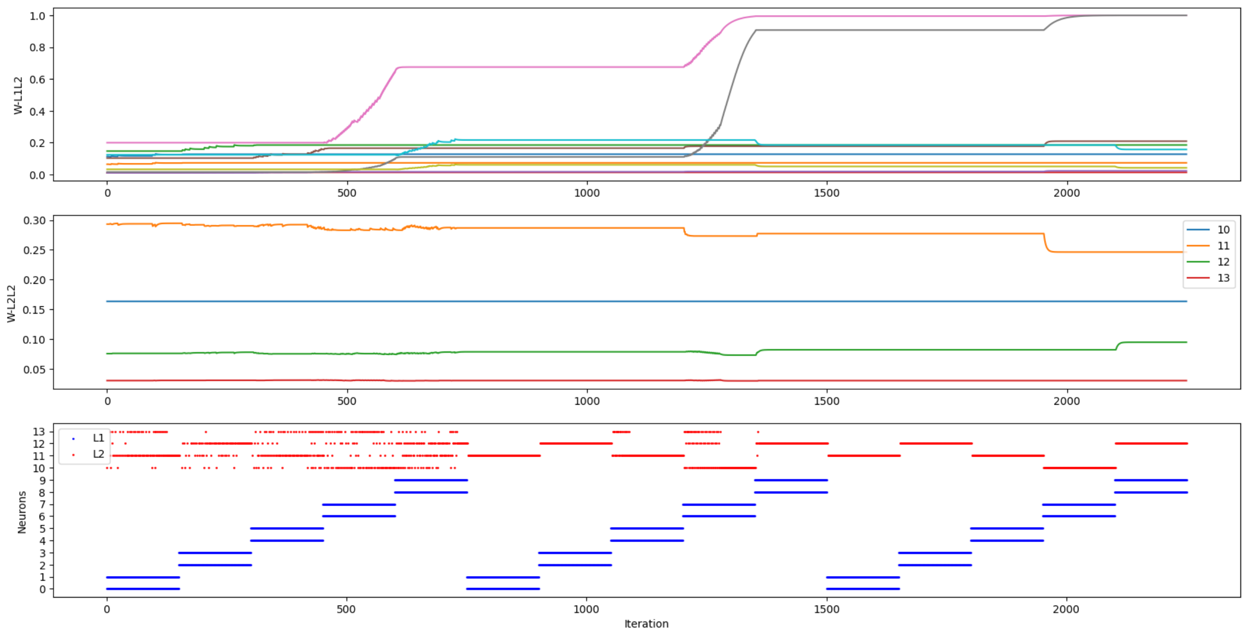

Figure 13 Network activity on distinct pattern with VBH mechanism added to layer 2 synapses. top panel: weight changes of excitatory synapses between input and output neurons, mid panel: weight changes of inhibitory synapses between output neuron. bottom panel: raster plot of 10 input and 4 output neurons activity with 5 distinct pattern input.

2. Effect of 2nd Layer Size

We reduce the number of output neurons to 4 and adjust the network slightly to get specific output for 5 distinct input patterns fed to the network. The result can be seen in **Figure 13**. The raster plot shows sparse spiking activity such that some neurons firing infrequently or not at all. In addition, weights show minimal changes over time and do not seem to adjust significantly. This indicates that the network is not capturing the complexity of the input patterns, suggesting that it has undergone ***underfitting***.

Then we increase the number of output neurons to 6 and make slight adjustments to the network to obtain specific outputs for 5 distinct input patterns. The results are shown in **Figure 14**. We can see that the pattern is almost distinguished at iteration 750, which is *one-third* of the iteration time needed for the same input with 5 neurons (**Figure 12**). Additionally, one of the neurons (neuron 12) does not participate in pattern recognition at all. Some weights converge too quickly to values that fit the training data, while others do not experience significant changes during the training process. This suggests that the network's complexity exceeds that of the data, leading to ***overfitting***.

3. Voltage-Based VS Activity-Based Homeostasis

`class` **`VoltageBaseHomeostasis`**`(Behavior)`

In each iteration, the implementation first identifies the over-threshold potentials; The expression `neurons.v > self.max_ta` evaluates to True (or 1) for neurons that their membrane potential exceeds the upper threshold (`self.max_ta`) and evaluates to False (or 0) otherwise, resulting in zero adjustment. The amount of difference is then calculated and stored in a variable named `greater`.

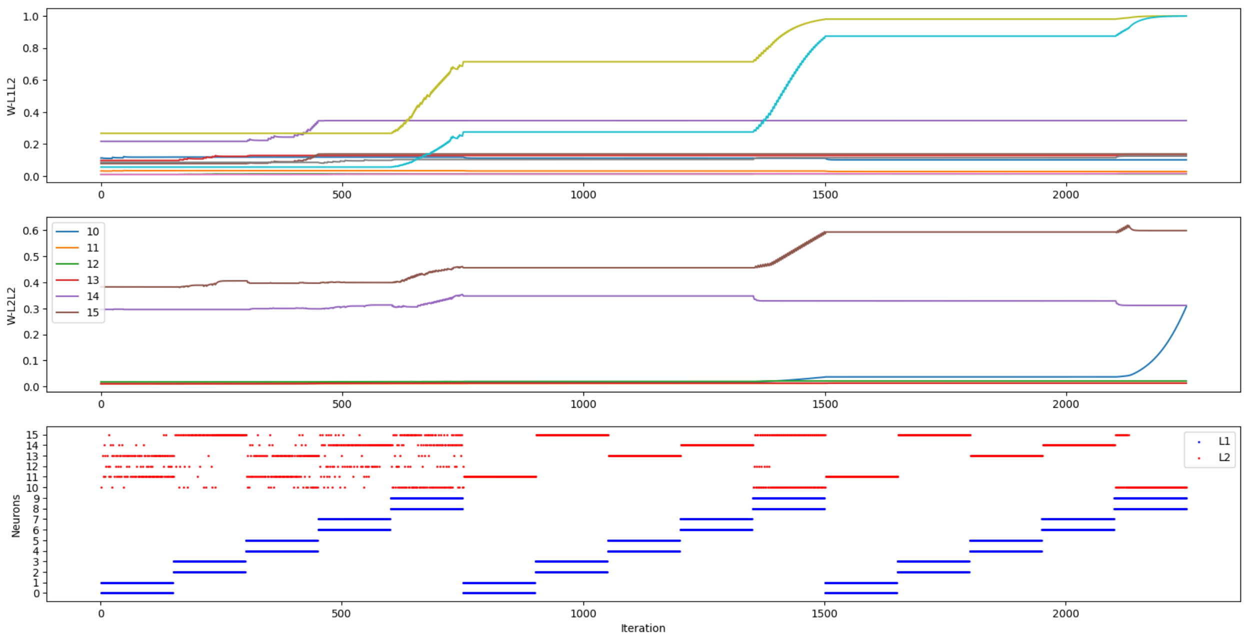

Figure 14 Network activity on distinct pattern with VBH mechanism added to layer 2 synapses. top panel: weight changes of excitatory synapses between input and output neurons, mid panel: weight changes of inhibitory synapses between output neuron. bottom panel: raster plot of 10 input and 6 output neurons activity with 5 distinct pattern input.

Then, the expression `neurons.v < self.min_ta` evaluates to True (or 1) for neurons that their membrane potential falls below the lower threshold (`self.min_ta`) and evaluates to False (or 0) otherwise, resulting in zero adjustment. The amount of difference is then calculated and stored in a variable named `smaller`.

The total adjustment for each neuron is then calculated and multiplied by `self.adj_strength` (or `eta_ip`) to scale the adjustment according to the desired strength. This values are added to a variable `neurons.exhaustion` which is initialized as zero at the start of the training process. This variable represents the accumulated adjustment needed to bring the membrane potential back within the desired range.

Finally, the accumulated adjustment is subtracted from the current membrane potential so the membrane potential gets closer to the desired range.

Why does it work?

By continuously adjusting the membrane potential based on deviations from the thresholds, the implementation ensures that neurons do not become overly active or inactive. This dynamic equilibrium helps maintain a balanced level of activity across the network, preventing runaway excitation or complete inactivity.

The gradual adjustments (controlled by `self.adj_strength`) prevent abrupt changes in neuronal activity, which could destabilize the network. This gradual approach allows the network to adapt smoothly to changes in input or internal dynamics, enhancing its robustness.

`class` **`ActivityBaseHomeostasis`**`(Behavior)`

In each iteration, the implementation adds `self.firing_reward` to its activity score if a neuron has fired and adds `self.non_firing_penalty` to its activity score if a neuron has not fired. These two parameters are initialized as 1 and a function of desired activity rate and the window size, respectively. Then the calculated activity changes are added to the `self.activities` vector, accumulating the activity scores over time. This vector is re-initialized at each `self.window_size` iterations.

At every `self.window_size` iterations, The accumulated activities are multiplied by `self.updating_rate` to determine the change in the firing threshold. The calculated change is subtracted from the neuron's current threshold, lowering the threshold for neurons that have been less active and raising it for those that have been more active. The `self.updating_rate` is multiplied by `self.decay_rate` to gradually reduce the rate of threshold adjustments over the course of iterations.

Why does it work?

The implementation dynamically adjusts the firing thresholds based on recent activity, ensuring that neurons maintain a balanced level of activity. Neurons that are too active will have their thresholds increased, making it harder for them to fire, while less active neurons will have their thresholds decreased, making it easier for them to fire. This is achieved by rewarding firing and penalizing non-firing and helps prevent neurons from becoming overly active or inactive, promoting a stable and efficient network.

The use of a window size allows the network to adapt its thresholds based on activity over a specified period, rather than making abrupt changes. Also, the decay of the updating rate ensures that the adjustments become more conservative over time, allowing the network to settle into a stable state.

A Comparison

The *Activity-Based Homeostasis* adjusts the firing thresholds based on their recent activity. This implementation directly influences the firing behavior of neurons by adjusting thresholds so it can be fine-tuned to achieve specific activity levels by adjusting reward and penalty values. But is requires careful tuning of parameters such as window size, reward, penalty, updating rate, and decay rate.

The *Voltage-Based Homeostasis* adjusts the membrane potentials of neurons based on deviations from predefined upper and lower thresholds. So it maintains a dynamic equilibrium by keeping membrane potentials within a specified range, preventing runaway excitation or complete inactivity. It is simpler to tune compared to activity-based homeostasis but it influences firing behavior indirectly by adjusting membrane potentials rather than firing thresholds so it can be less precise in achieving specific activity levels.

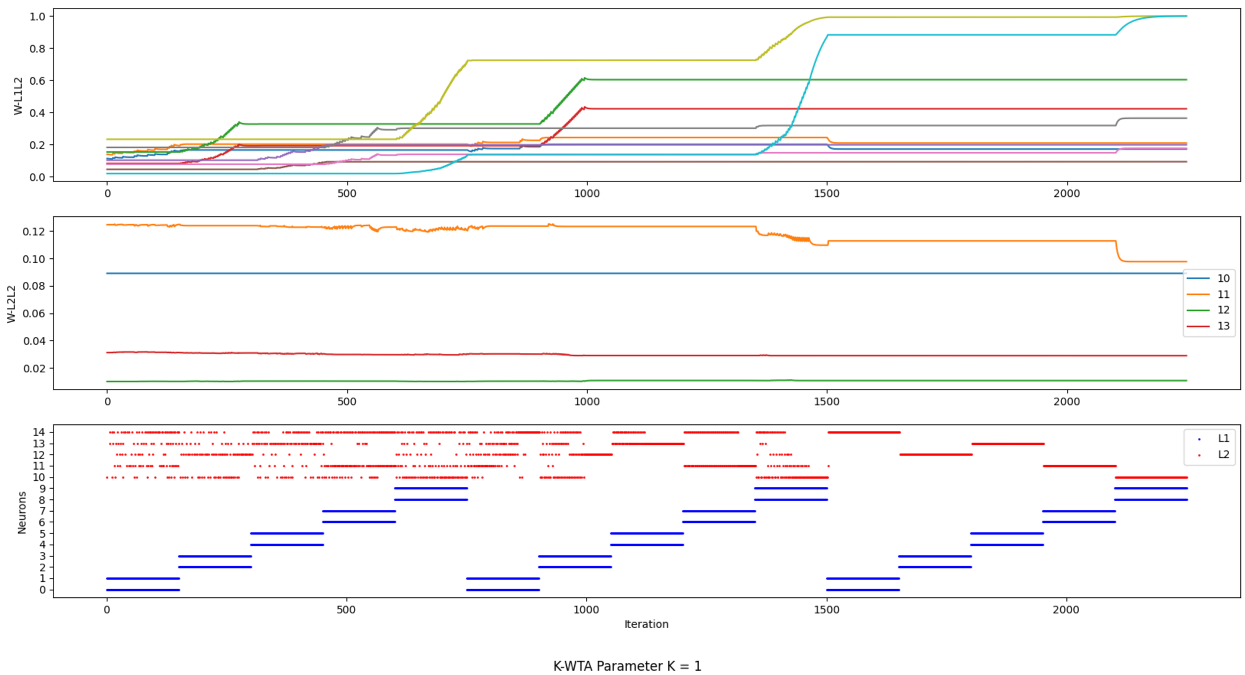

Figure 15 K-WTA parameter K tuning.

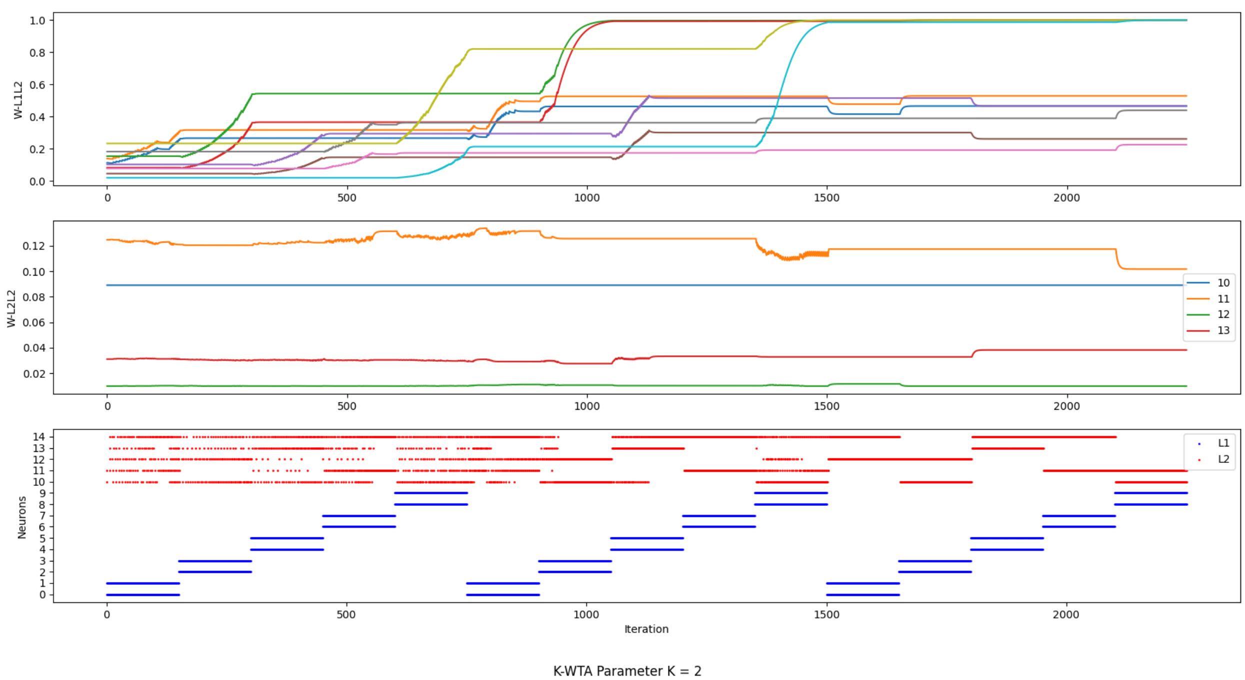

Figure 16 K-WTA parameter K tuning.

Task 5: Parameter Tuning

K-WTA Parameter K

Variations of the K parameter are shown in **Figures 15**, **16**, and **17**. When K=1, the network chooses only one output neuron to spike in each iteration. This helps assign each pattern to a

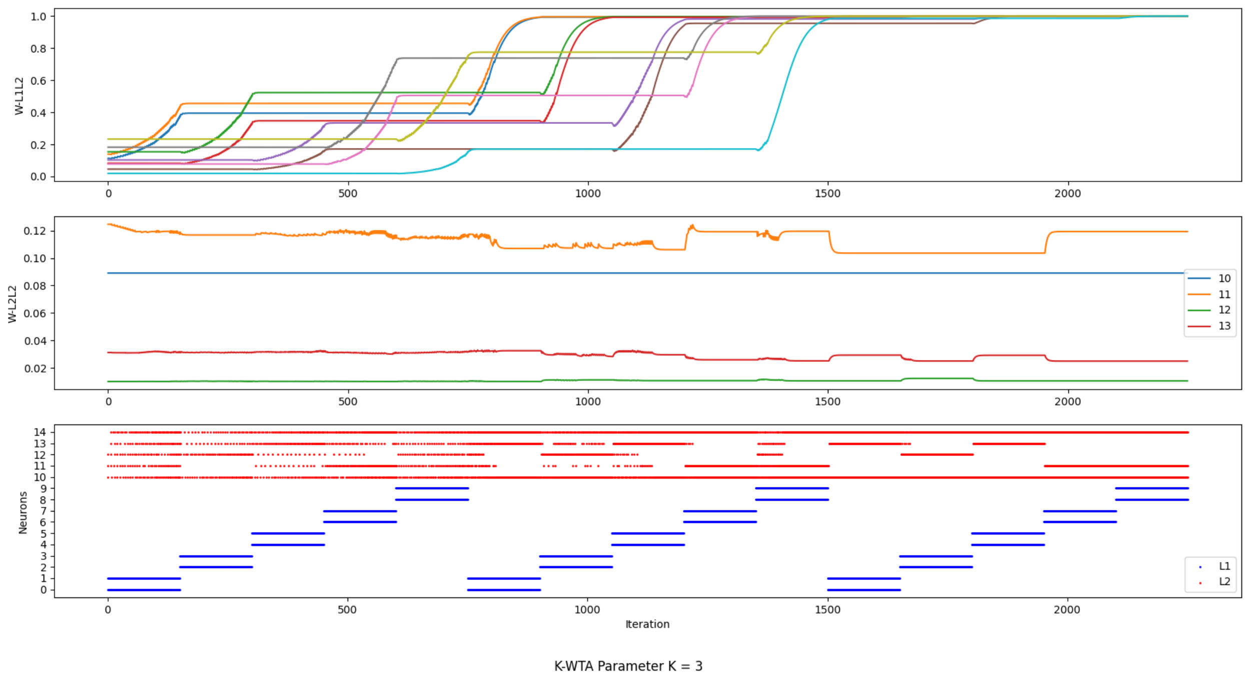

Figure 17 K-WTA parameter K tuning.

specific output neuron while inhibiting the activity of other neurons (**Figure 15**). When K=2, we observe increased activity from the output neurons during the training process. By the end of the training, two neurons activate for each input pattern, which is expected since we allowed two neurons to activate in each iteration (**Figure 15**, bottom panel). Similarly, when K=3, three neurons activate for each output pattern, and the inhibition effect is further decreased (**Figure 17**).

When K is small, many neurons are suppressed and do not update their weights, leading to more stable weights. As K increases, the suppression effect is reduced, and more neurons are actively updating their weights.

With a higher K, more neurons are competing to be among the top K winners. This increased competition can lead to more dynamic changes in weights as neurons adjust to outcompete others. As K increases, a larger subset of neurons is involved in processing diverse input patterns. More active neurons mean more frequent weight updates, contributing to greater fluctuation.

Voltage-Based Homeostasis Parameter η

This parameter controls the updating speed of the homeostasis homeostatic process updates the neuron's membrane potential to maintain stability. Variation of this parameter can be seen in **Figures 18**, **19**, and **20**. When η=001, the the network gradually applies the voltage control and can accurately capture each input pattern (**Figure 18**).

A higher η value means that the neuron's membrane potential will adjust more quickly in response to deviations from the desired homeostatic level so neurons will be more responsive to changes in input, quickly adapting their membrane potentials to maintain stability. This rapid changes can lead to erratic updates in synaptic weights which is seen in **Figure 19**, top panel. This erratic behavior disrupts the learning process, causing the network to struggle in finding a stable

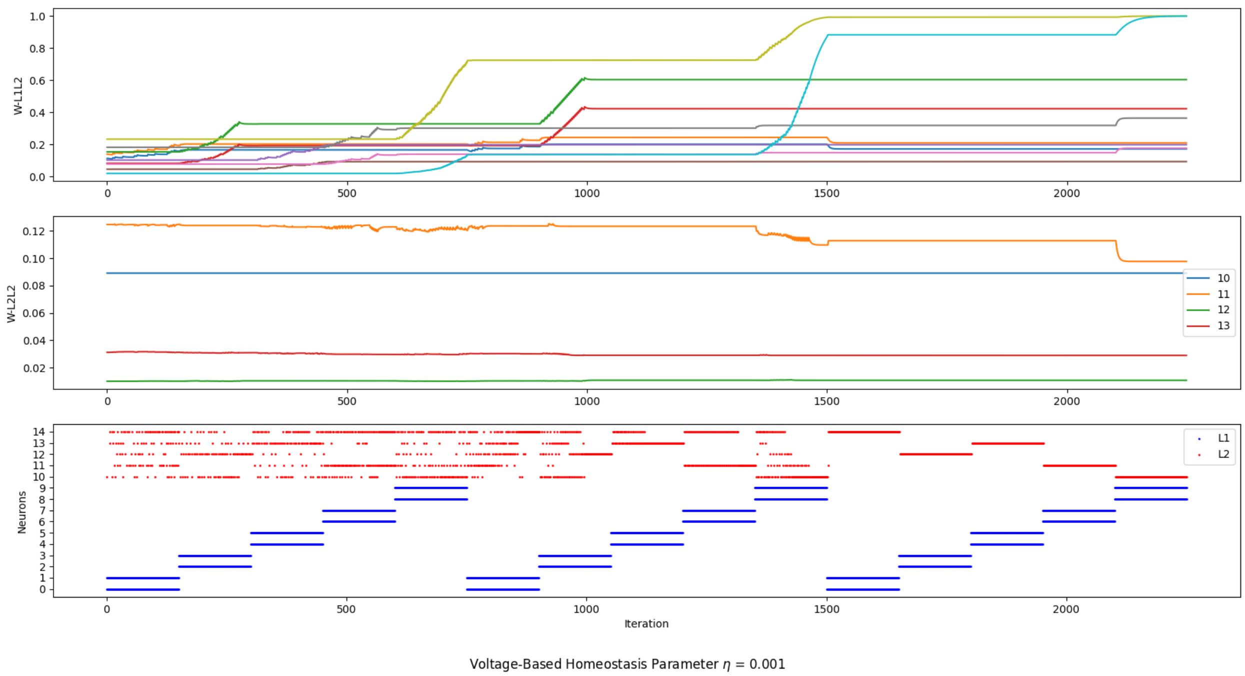

Figure 18 VBH parameter η tuning.

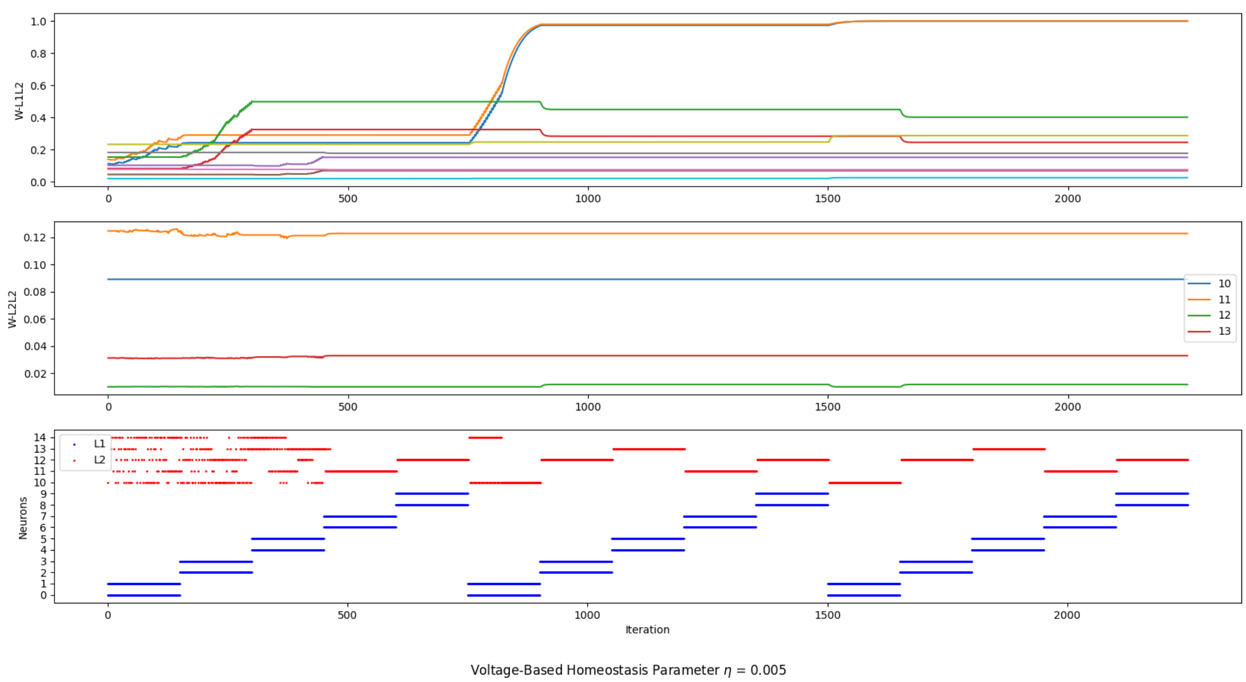

Figure 19 VBH parameter η tuning.

and accurate representation of the input data so the accuracy of the network's output is degraded which is what we see in **Figure 19**, bottom panel

A lower η value means that the neuron's membrane potential will adjust more slowly, taking longer to reach the desired homeostatic level so neurons will be less responsive to changes in input, with slower adaptations in their membrane potentials making it harder for the network to adapt to

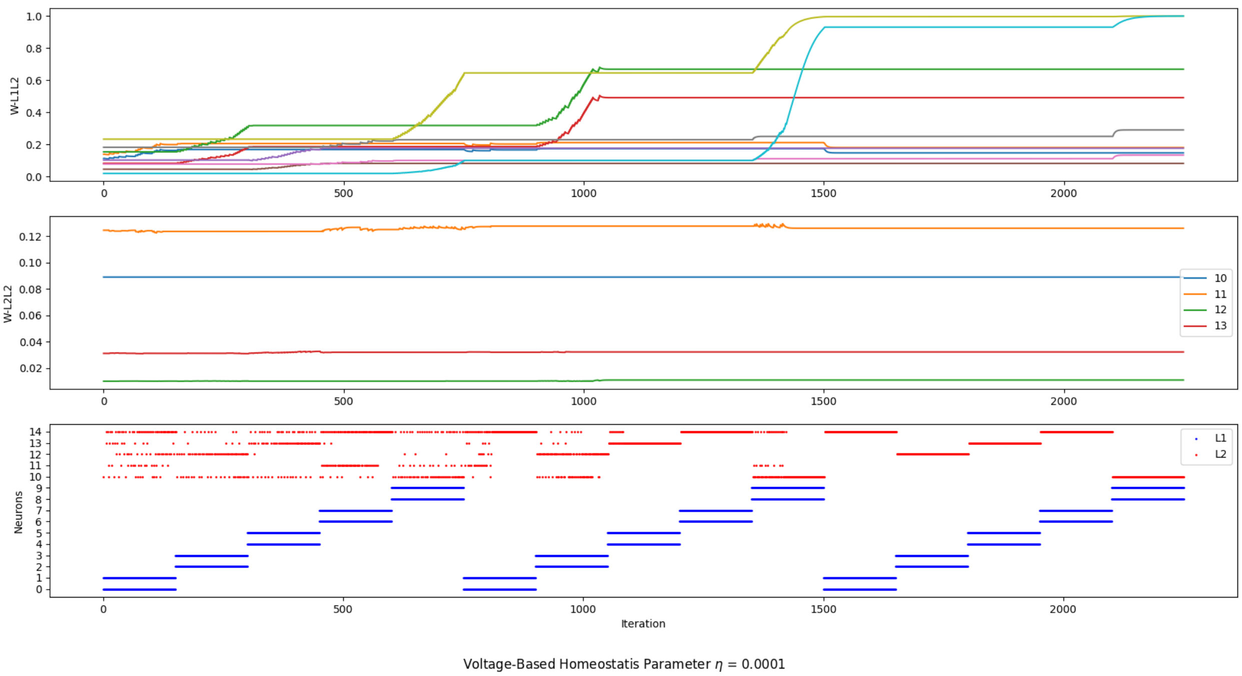

Figure 20 VBH parameter η tuning.

new patterns or variations in the input data. The neurons in network shown in **Figure 20** struggle to maintain homeostasis so they have been overly suppressed over time leading to inaccurate results.

Spike Trace Parameter τ

The spike trace parameter τ represents the time constant of the exponential decay of the post-synaptic potential (PSP) or the synaptic trace in a spiking neuron model. τ determines how long the effect of a spike persists in the neuron's membrane potential. A larger τ means the effect of a spike lasts longer, allowing the neuron to integrate information over a longer period. Variations of this parameter is shown in **Figures 21, 22** and **23**.

With a higher τ, neurons integrate spikes over a longer period, which can lead to a higher likelihood of the membrane potential remaining above the threshold for longer durations. However, this can also mean that neurons are less likely to fire frequently because the membrane potential does not reset quickly. The prolonged effect of each spike can cause the neuron's membrane potential to saturate, reducing the overall firing rate as the neuron spends more time near-threshold state which explains the low activity of **Figure 22** and **23**, bottom panel compared to **Figure 21.**

Also, weight fluctuations seem to increase and weight adjustments are faster with higher τ value. This is because each spike influences the neuron's activity for a longer period. This extended influence can lead to more significant changes in synaptic weights during STDP learning. These amplified synaptic changes can accelerate the learning process, allowing the network to adjust its weights more quickly and converge faster which can also be observed in the figures.

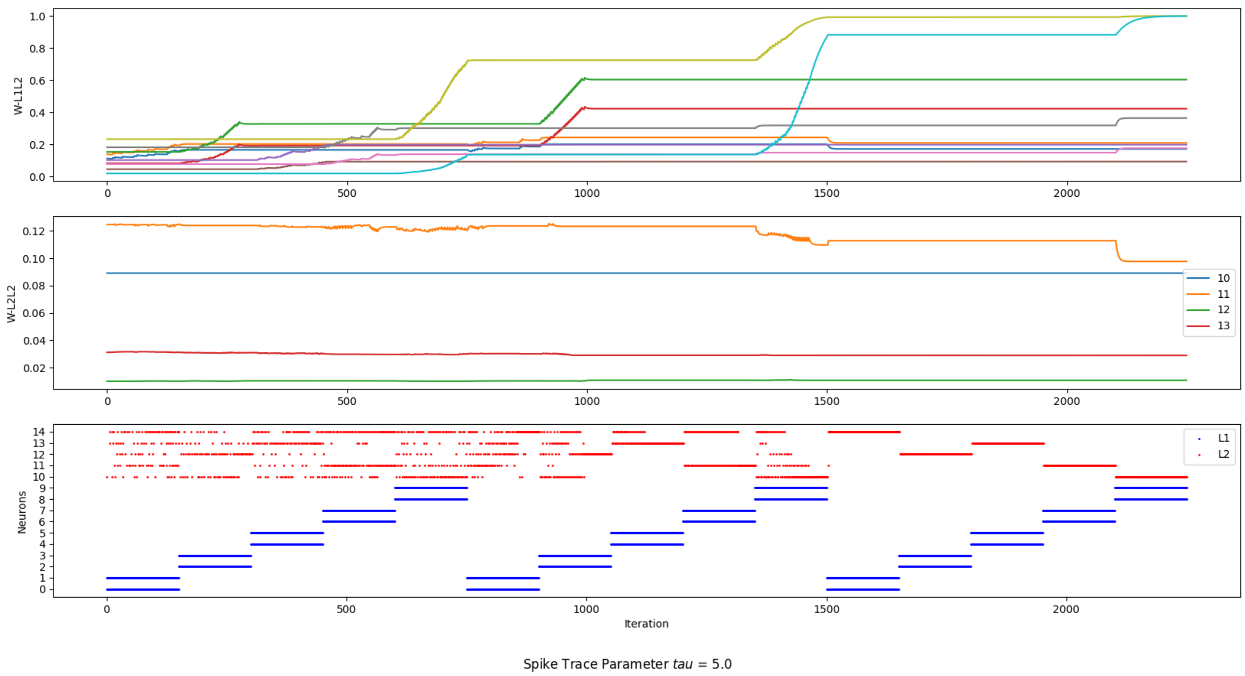

Figure 21 Spike trace parameter τ tuning.

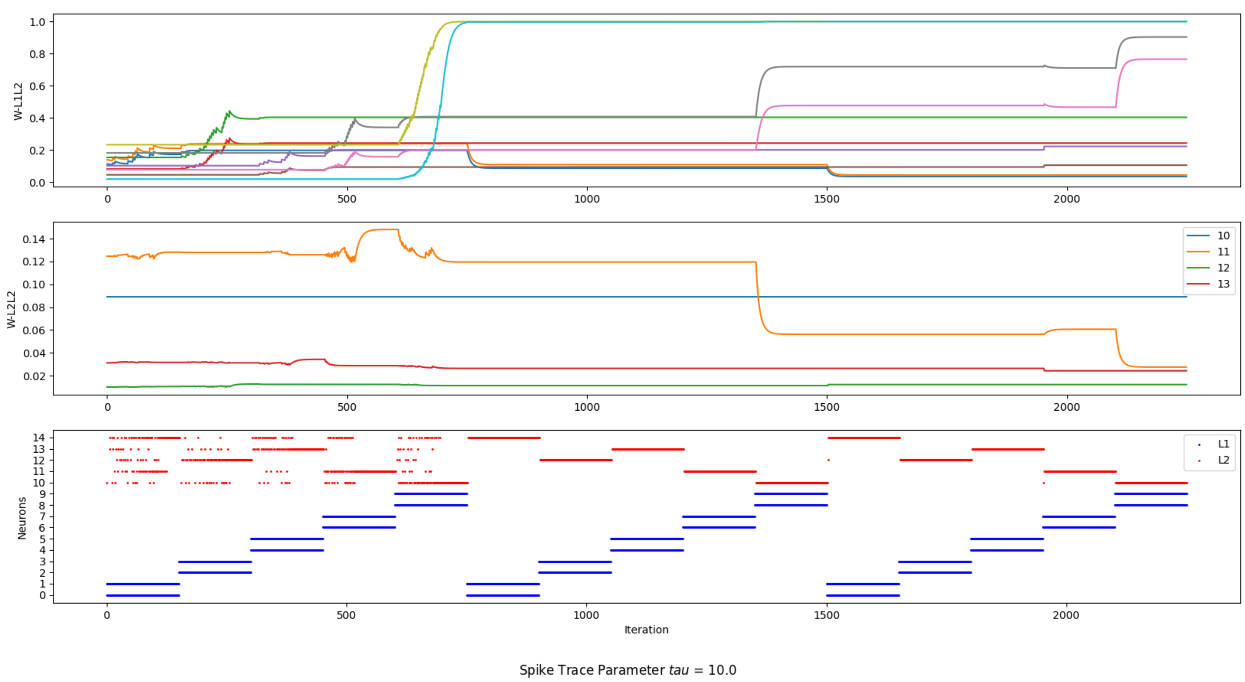

Figure 22 Spike trace parameter τ tuning.

A very high τ value means that the network integrates information over an excessively long period. This can cause the network to lose the ability to resolve fine temporal details in the input data and it becomes less responsive to new input patterns due to the slow decay of PSP, which explains the poor performance seen in **Figure 23**.

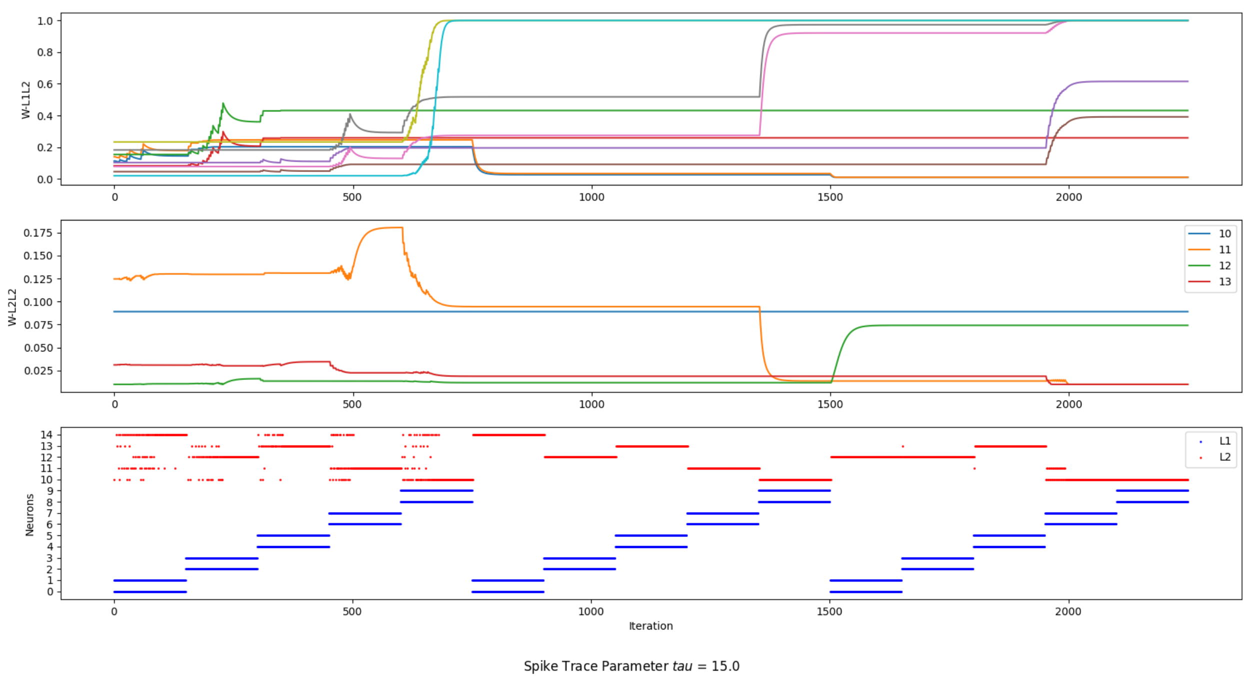

Figure 23 Spike trace parameter τ tuning.

Synaptic Weight Scaling Factor (SWF)

The parameter SWF acts as a scalar multiplier for the synaptic weights. The variations of this parameter for excitatory synapses of the network can be seen from **Figure 25, 26** and **27.** There are a few key points about what the SWF actually does for each of excitatory and inhibitory synapses:

1. **Role:**

   - *Input to Output Synapses (Excitatory):* When SWF is increased for the synapses connecting input neurons to output neurons, it directly amplifies the excitatory input signals. This leads to stronger post-synaptic potentials (PSPs) in the output neurons, causing significant changes in synaptic weights as the network adjusts to the heightened input.

   - *Output to Output Synapses (Inhibitory):* Increasing SWF for the synapses among output neurons primarily affects the inhibitory interactions. These synapses are responsible for lateral inhibition, which regulates the activity of neighboring neurons to maintain balance and prevent excessive firing.

2. **Impact on Synaptic Plasticity**

   - *Excitatory Synapses (Input to Output):* The increased SWF for excitatory synapses results in more pronounced synaptic plasticity. The stronger excitatory inputs lead to larger adjustments in synaptic weights as the network learns to respond to the input patterns.

   - *Inhibitory Synapses (Output to Output):* The increased SWF for inhibitory synapses enhances the lateral inhibition among output neurons. However, this does *not* directly contribute to synaptic plasticity in the same way as excitatory inputs. Instead, it modulates the overall activity levels and competition among output neurons.

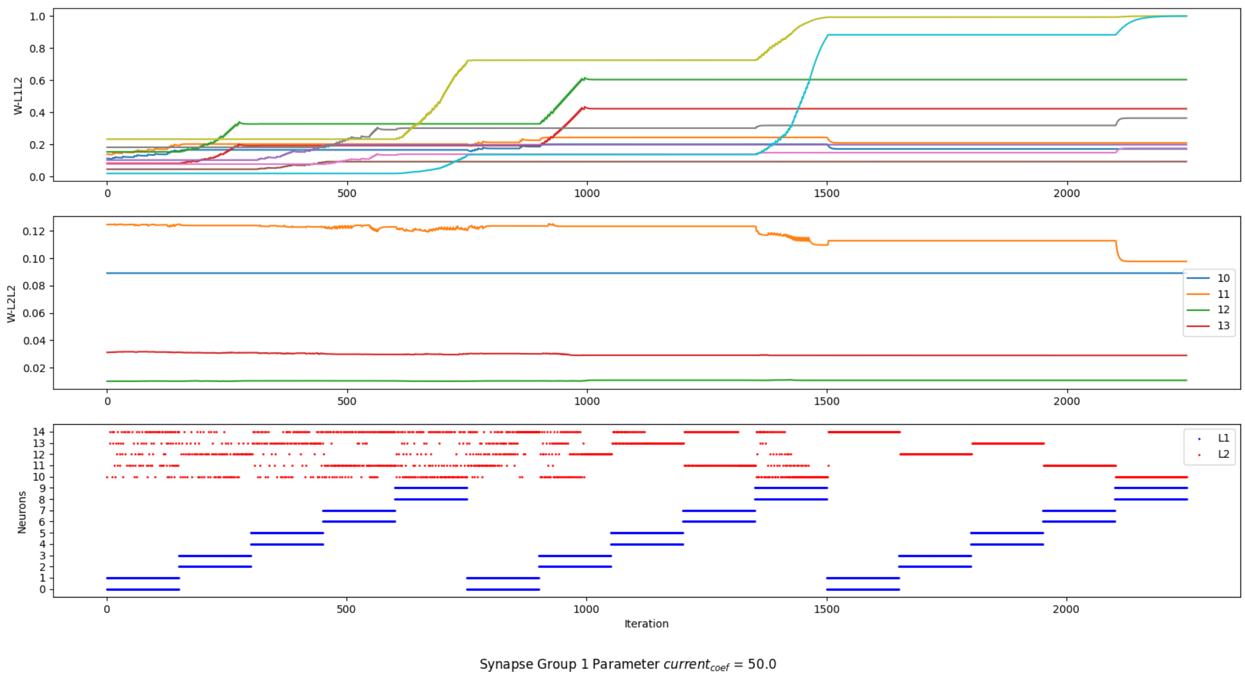

Figure 24 SWF tuning (excitatory).

3. **Mechanisms of Weight Change**

- *Excitatory Weight Changes:* The synaptic weights between input and output neurons are more susceptible to changes because they are directly involved in encoding the input patterns. The increased SWF amplifies the learning signals, leading to significant weight adjustments.
- *Inhibitory Weight Changes:* The synaptic weights among output neurons are involved in maintaining balance and preventing runaway excitation. The increased SWF for these inhibitory synapses enhances their regulatory role but does not directly drive learning-related weight changes. Also, the increased lateral inhibition helps to stabilize the network by preventing excessive firing of output neurons. This stabilization effect results in less variability and smaller weight changes compared to the situation where excitatory inputs are amplified.

Increasing this coefficient for *excitatory* synapses amplifies the strength of these synapses which is why we see an increase in synaptic weights between input and output neurons in **Figures 25**, **26** and **27** top panel. The amplified synaptic strength accelerates the learning process by making the network more responsive to input patterns. This heightened responsiveness leads to faster convergence as the network quickly adjusts its weights to minimize error (**Figures 25**, **26** and **27** bottom panel).

With very high value for SWF some output neurons (specifically neuron 11 and 12 in **Figure 26** bottom panel) become unresponsive during the training process. We already mentioned that high value for this parameter amplifies the synaptic weights significantly, leading to extremely strong excitatory inputs to the output neurons. Neurons that receive slightly stronger inputs can dominate the KWTA competition, consistently winning and inhibiting the other neurons. This can lead to a situation where only a few neurons are active, while others remain suppressed and unresponsive. Also, the high excitatory weights can lead to excessive activation of the inhibitory mechanisms. The lateral inhibition, which is supposed to balance the activity among neurons, may become too strong, suppressing the activity of some neurons entirely.

In addition, Homeostasis aims to maintain a balanced level of activity across neurons. However, with very high SWF, the homeostatic adjustments fails to bring the activity of the highly excited neurons down to a normal level, leading to persistent unresponsiveness in other neurons.

A very low SWF diminishes the influence of synaptic weights, leading to weaker signal transmission between neurons. Insufficient synaptic strength can lead to under-excitation, where neurons do not receive enough input to fire so the network becomes less responsive to input patterns, leading to slow learning and poor performance seen in **Figure 25**. Also, A very low SWF diminishes the influence of synaptic weights, leading to weaker signal transmission between neurons so the inhibitory effect is reduced, leading to less effective lateral inhibition. The reduced inhibition allows more neurons to spike simultaneously, which can disrupt the network's ability to learn and generalize effectively.

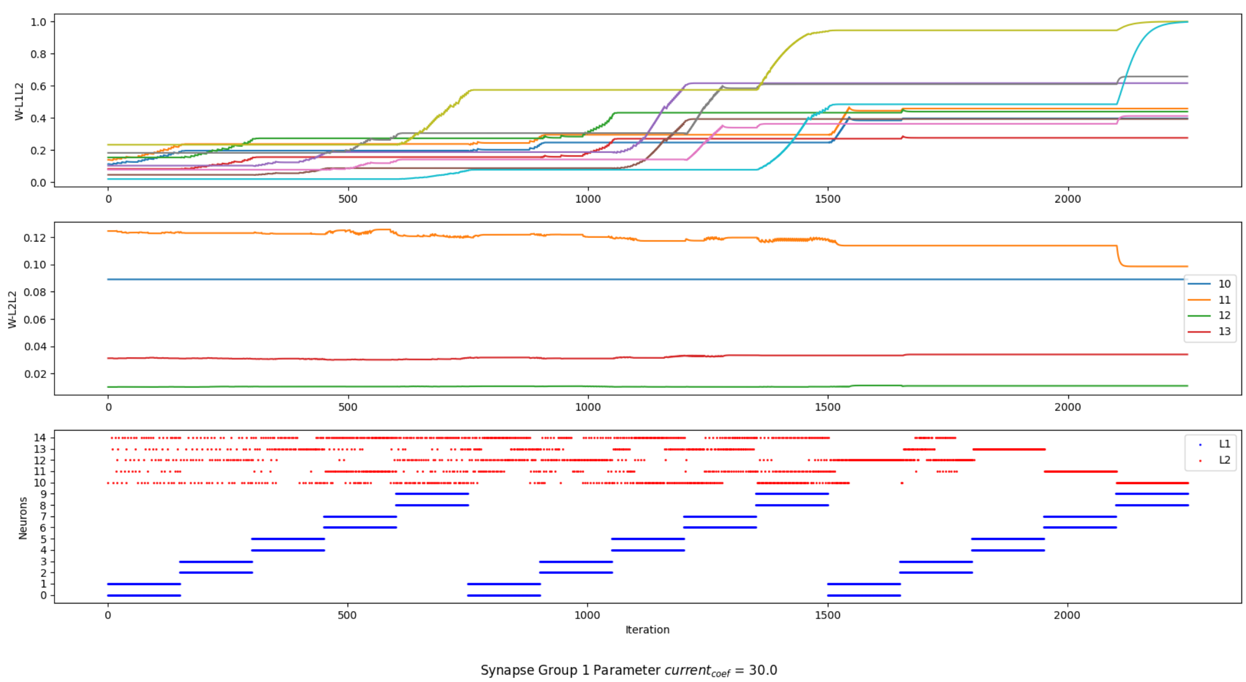

Figure 25 SWF tuning (excitatory).

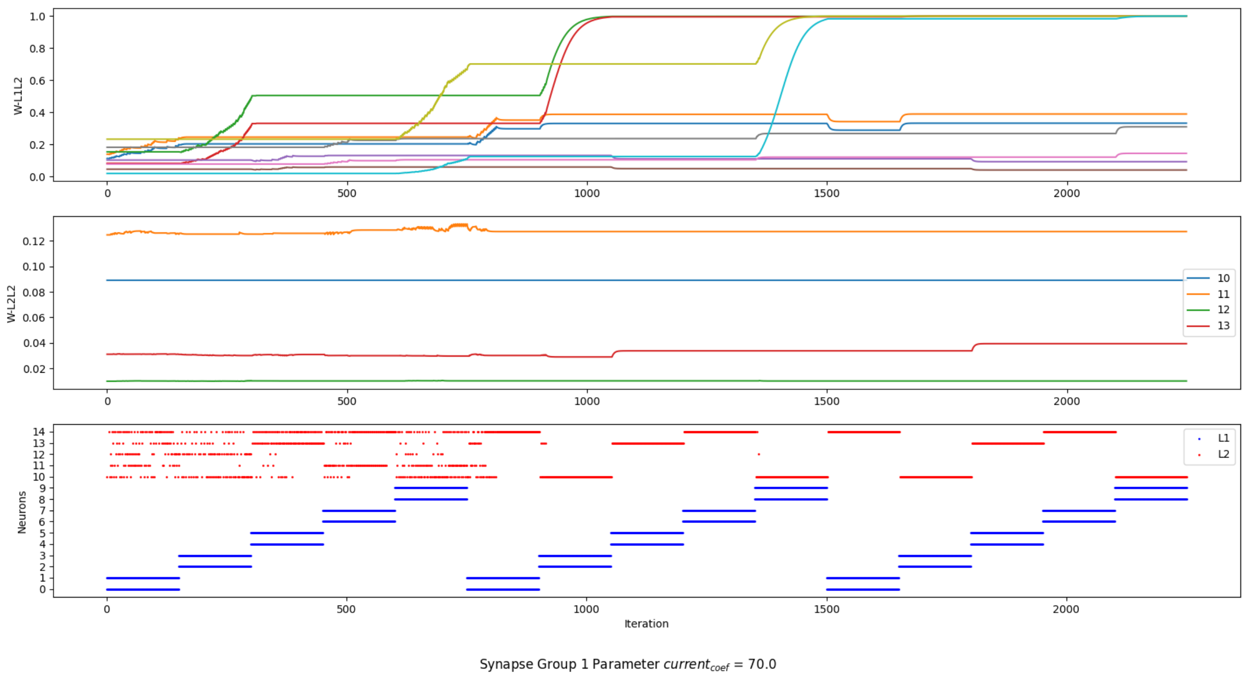

Figure 26 SWF tuning (excitatory).