Supplementary Material

Hammouda Elbez [1,*] **and Mazdak Fatahi** [1]

[1] *Univ. Lille, CNRS, Centrale Lille, UMR 9189 - CRIStAL - F-59000 Lille, France*

Correspondence*:
Hammouda Elbez
hammouda.elbez@univ-lille.fr

1 SPIKES ACTIVITY PER NEURON FOR EACH LAYER

In this section, we observe the spikes activity per neuron in each layer when using the baseline and the compressed STDP-based network. The goal is to analyze the effect of the compression on each neuron's spikes activity during the learning phase. For this experiment, we use two convolutional-pooling pairs and SVM to classify the Caltech face/motorbike dataset (400 train images). The first convolutional layer contains 32 neurons, and the second 64 neurons. For the compressed network, we record the following rates: 49% for the first layer and 52% for the second layer. Moreover, we train each layer ten epochs and report the recorded accumulated spikes activity in Figure 1.

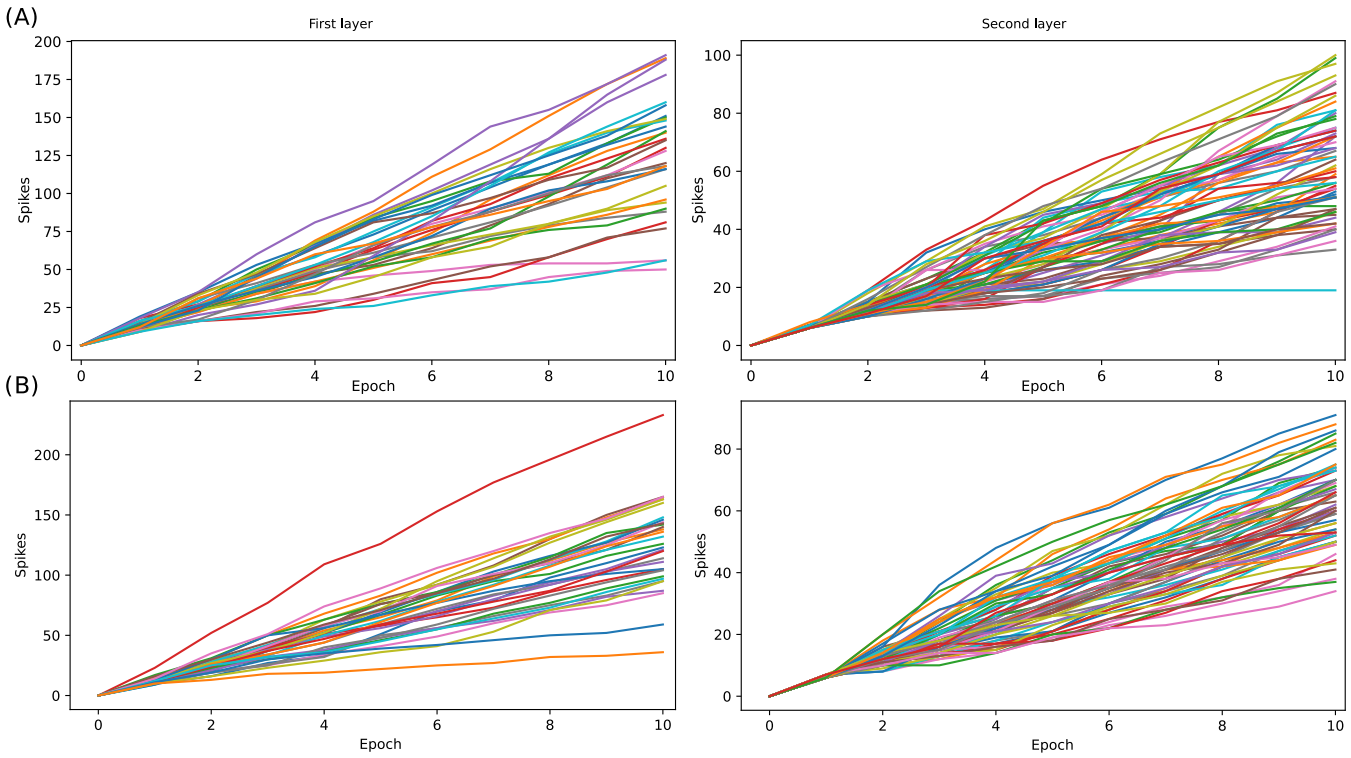


Figure 1. Accumulated spikes activity per neuron for the first and second layer, for (A) the baseline and (B) the compressed STDP-based network.

We can observe from Figure 1 that the spikes activity in all the figures is increasing during training. Moreover, if we compare the activity in both cases (baseline and compressed), we see that the spikes activity is different. In the first layer, we have values between 50 and 190 for the baseline and 36 and 233 for the

compressed network due to compression. However, it is interesting to see that the maximum value increases in the compressed network due to the threshold adaptation mechanism used in the network. For the second layer, we got spikes values between 19 and 100 for the baseline and 37 and 88 for the compressed network. Therefore, for the second layer, we see that the maximum spikes per neuron value are less than the first layer in the baseline and the compressed one. Moreover, the second layer of the compressed network has more active neurons, which may be the result of the threshold adaptation mechanism and the network trying to maintain the spike activity in the layer after reducing the synapses.

2 NEURONS THRESHOLD EVOLUTION ON THE CSNN-SIMULATOR

Due to the importance of the threshold adaptation mechanism used in the STDP-based network, we analyze the evolution of the neuron's threshold during training in this section. We keep the same network architecture with the same compressed rate for this experiment and use the same dataset. In Figure 2, we can see the activity of the two layers for both networks.

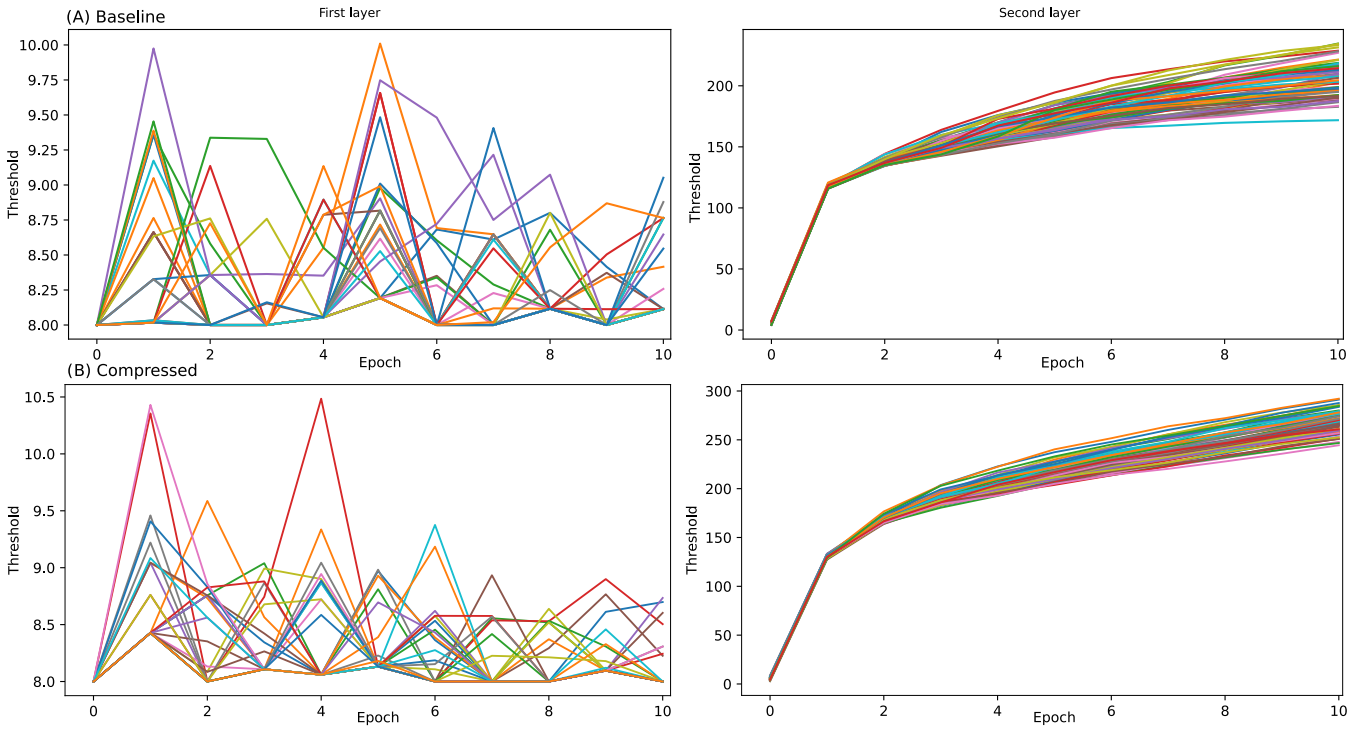


Figure 2. Neurons threshold evolution of the first and second layer during training, for (A) the baseline and (B) the compressed STDP-based network.

Interestingly, the neuron's threshold dynamic in the first layer differs from the second in both use cases because the first layer is exposed to the inputs directly after preprocessing. In contrast, the second layer receives data processed by the first convolutional and pooling layer. In the simulation, there is a th_{min} value, which represents the minimum possible threshold value, and we set it to 8 for the first layer and 1 for the second layer. Comparing the first layer in both cases, we see that the threshold values increase during training. In the baseline, the maximum value is ten, and in the compressed one, it is slightly higher. Furthermore, after epoch 4, we can see that in the compressed network, the thresholds are lower and stable between 8 and 9.4. Yet, in the baseline, the values kept changing the same way. In the second layer, we can see that the neuron's threshold is much higher after ten epochs than in the first layer. In the baseline, the

values are between 171 and 234. In the compressed one, the values are between 244 and 292. Therefore, the increase in the neuron's threshold value for the compressed second layer is a reaction of the threshold adaptation mechanism to preserve the required activity.

3 WEIGHTS DISTRIBUTION IN CSNN-SIMULATOR AND SPINNAKER

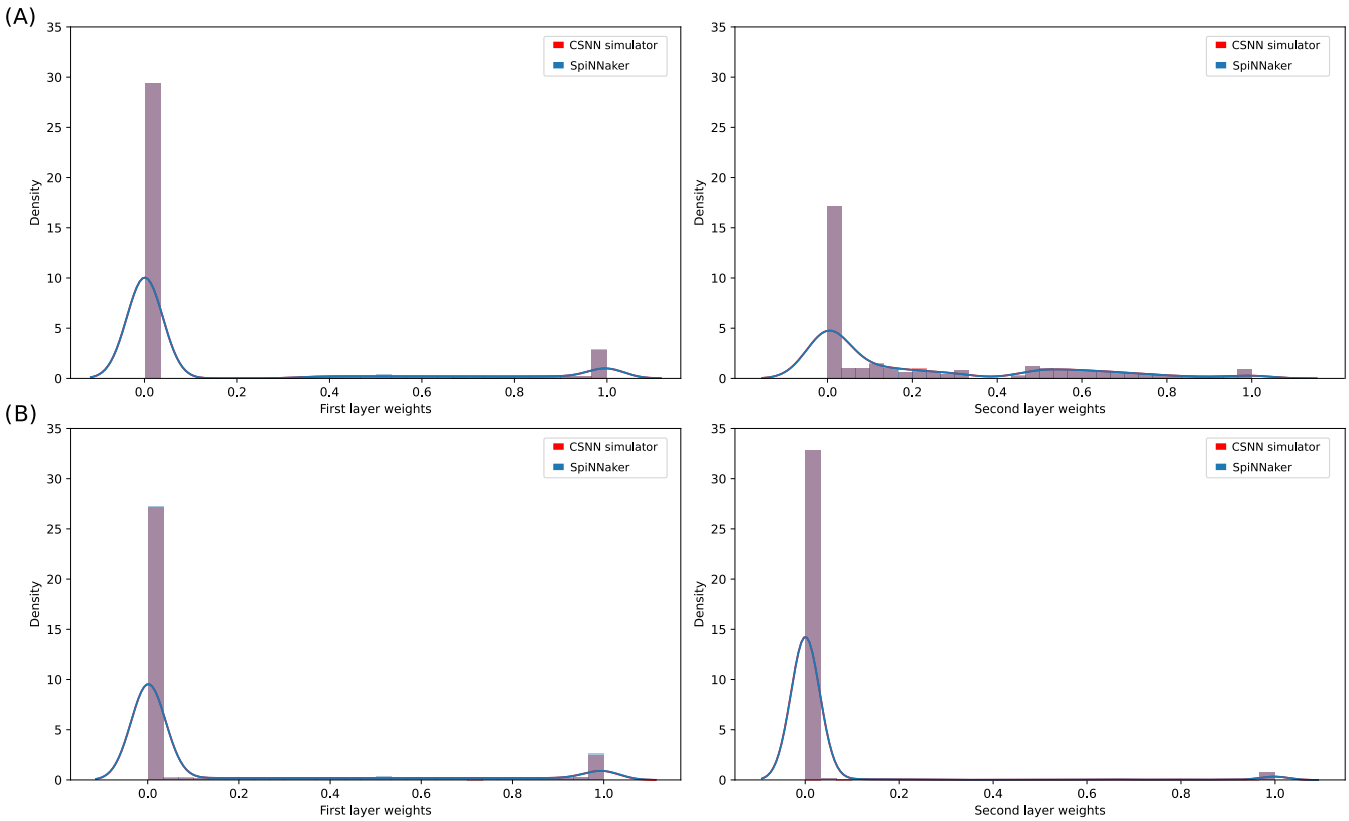


Figure 3. Trained weights distribution in csnn-simulator and SpiNNaker for (A) the baseline and (B) the compressed STDP-based network.

In this section, a network of two fully-connected layers of 50 and 128 neurons is trained with the csnn-simulator. Following that, we upload the trained weights on the SpiNNaker board. Figure 3 and Table 1 show the weight distribution for the two use cases in both layers.

Table 1. Weights distribution in csnn-simulator and SpiNNaker

		Baseline			Compressed		
	Weights	0	$0 < w < 1$	1	0	$0 < w < 1$	1
csnn-simulator	L1	31970	4465	2765	27348	10680	1172
	L2	2904	3340	156	5772	505	123
SpiNNaker	L1	31970	4335	2895	29911	6843	2446
	L2	2917	3327	156	5816	457	127

According to Table 1 in the first layer of the baseline, a considerable amount of the weights are zero, and the rest is distributed between zero and one. In the compressed network, we have more weights between zero and one and less at zero and one. Most of the weights are between zero and one for the second layer of the baseline. Moreover, the weights are mostly zeros for the compressed network, which is the expected result after compression. For the SpiNNaker board, we extract the weights from the board after uploading them to see if they are unchanged. As shown in Figure 3, the extracted weights are in the same distribution. However, since ARM968 has no specific hardware for floating points, SpiNNaker supports up to 32-bit fixed-point precision. Therefore, the stored weights in SpiNNaker can be slightly different.

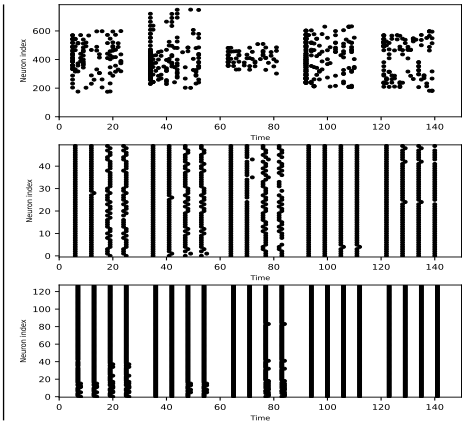
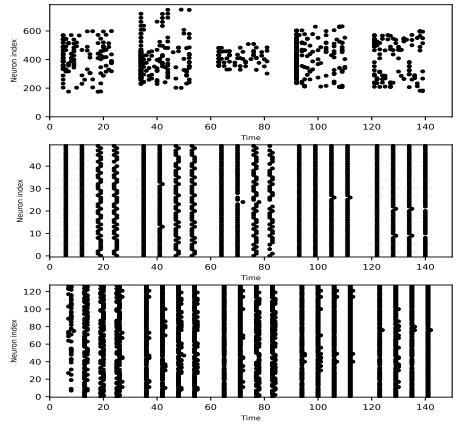
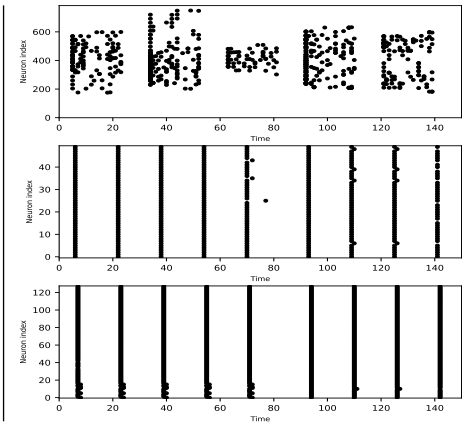
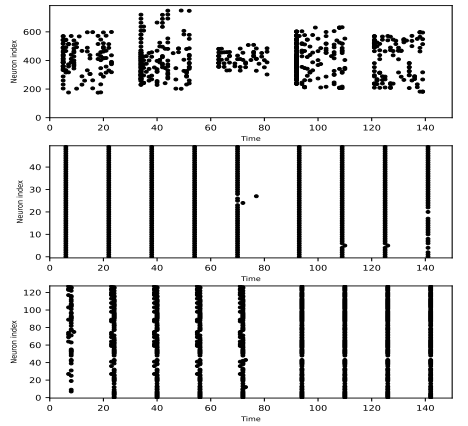
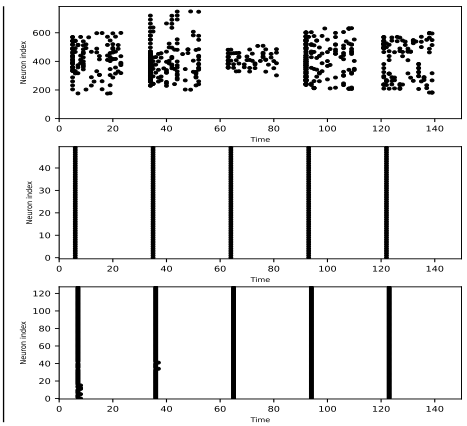
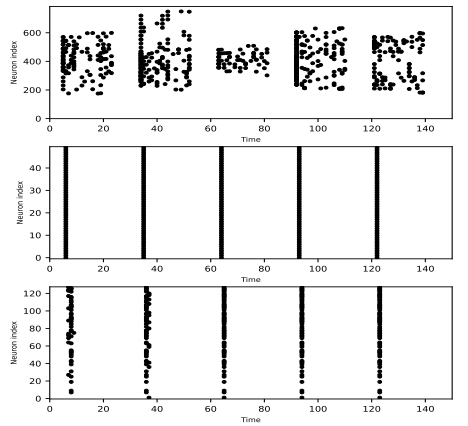
4 REDUCING THE SPIKES ACTIVITY IN SPINNAKER

As we saw in the manuscript, the SpiNNaker board generates many spikes, which may affect the classification task. Therefore, there are different possibilities to reduce it. This section discusses he use of neuron parameter tuning.

We can prevent the neurons from generating more spikes by tuning the neural parameters. In Table 2, we test different values of the refractory period. In our experiment, we present four samples, each for 20ms, and the resting time is 10ms. Therefore, the refractory period τ_{refrac} in our experiment equals 5ms, 15ms, 25ms, and 35ms.

As shown in Table 2, by increasing τ_{refrac} we have fewer spikes, and still, the neurons are triggered with the corresponding inputs. However, as shown in the same table, higher values of τ_{refrac} affect the network responses, as seen in the number of spikes of the two layers. As we mentioned, other parameters (v_{th}, v_{rest}, v_{reset}) also have an effect on the neuron activity, which will be explored in future works.

Table 2. Spikes activity per layer with different refractory period (5ms and 15ms)

Refractory period	Baseline	Compressed
5ms		
15ms		
25ms		
35ms