CSNN – An Exploration of Convolutional-Spiking Neural Networks with Varying Neuron and Synapse Parameters

Jakeb Chouinard
20826100

Abstract

This paper intends to explore how the introduction of biological features to an artificial Convolutional Neural Network impacts its classification performance. By transitioning from conventional Artificial Neurons to Spiking Neurons, we intend to see how adaptable a Convolutional Neural Network is to a Spiking Neural Network. This includes an exploration of linear and non-linear neuron spiking models as well as varying synaptic time constant values. To implement the Spiking Neural Network, we utilized the NengoDL conversion functionality on a Keras Convolutional Neural Network. We trained the Keras model on the CIFAR-10 dataset using the Adam optimizer and loaded the connection weights prior to model conversion. Each converted model was evaluated on 1000 testing images from the CIFAR-10 dataset and both the Categorical Accuracy and Categorical Cross-Entropy were recorded for each converted network. Replacements of linear neuron models, such as Rectified Linear Rate and Spiking Rectified Linear, with non-linear neuron models, such as Leaky Integrate-and-Fire and Adaptive Leaky Integrate-and-Fire, resulted in more robust classification in time-dependent models. Additionally, increases to the synaptic time constant resulted in the decrease of overall classification accuracy. The results of this exploration indicate that Spiking Neural Networks are a viable alternative to conventional Convolutional Neural Networks. Future research may be motivated by increasing the accuracy of the Spiking Neural Network post-conversion and requires additional computation resources.

Introduction

The Neocognitron was introduced in 1980 as a biologically-inspired neural network model and has since evolved into Convolutional Neural Networks (CNNs) [1]. While CNNs have been able to demonstrate image classification abilities above that of typical human observers, such as in the case of lesion detection where CNNs were capable of out-performing human observers in all cases except that of high-noise correlation [2], CNN implementations have tended away from their biologically-inspired origins. This is most obvious in the cases of neuron activation functions in Artificial Neural Networks and the use of techniques like maximum pooling. Most apparently, neurons do not output some constant, positive rate value proportional to its inputs; they fire at some rate dictated by the dynamics equations that govern its membrane potential [3]. This divergence in CNN implementation and biological neuron function inspires an examination of CNN effectiveness when non-biological functionality is replaced by biological analogues. This includes the conversion of the network to be temporal, the replacement of activation functions with neuron spiking models, and the varying of synaptic time constants.

Some other studies have been conducted to explore what happens when a conventional CNN is modified to have more biological behaviours, such as [4] where the introduction of a modest synaptic time constant ($\tau_{SYN}=0.001s$) and the replacement of ReLU-activated neurons with Rectified Linear Spiking neurons were explored. This study examined the accuracy of a

Keras-based CNN converted into differently parameterized Nengo-based Spiking Neural Networks (SNNs) and compared their accuracies across MNIST and P300 datasets. This study found that the converted SNNs were on average 2.67% less accurate than their CNN equivalents for the MNIST dataset and 5.60% less accurate for the P300 dataset. While this study makes interesting progress towards analyzing SNNs and their capabilities, the utilization of a linear spiking neuron model retains some of the non-biological concerns, as biological neurons exhibit typically non-linear (and potentially habituating) responses to their inputs. Additionally, the study only explored a single synapse time constant and assumed a maximum firing rate of 1000*Hz*. These values are not typical of neurons in the primary visual cortex (V1), as research indicates that V1 neurons fire around a nominal maximum value of 415*Hz* [5].

Another study exploring the modification of CNNs to have more biological aspects is [6]. This study similarly utilized a Keras-based CNN model and converted it into Nengo-based SNN models with varying parameters. This study used varying synapse time constant values as well as varying scaling for the firing rate. While this study was largely focused on energy consumption comparisons between existing models and their proposed model, it was found that the proposed model exhibited classification ability that was consistently better than the other models of interest, reaching 0.20% higher accuracy on Fashion-MNIST and 0.13% higher accuracy on MNIST when compared to other leading models. This study, similar to the previous, utilizes the Rectified Linear Spiking neuron model, lending to some biological inaccuracy. Unlike the previous study, this study does more adequate job of exploring the model space by varying both the synaptic time constant and the scaling factors. The results of the study indicate that high scaling factors lend to better overall accuracy with some asymptotic maximum accuracy that it approaches. This makes sense, as some infinitely high maximum firing rate would effectively result in neurons that were always on or always off, similar to the Artificial Neural Network implementation.

One upside of both previously-explored studies is their use of synaptic time constants that are typical of the biological range of values. Based on a list of Neurotransmitter Time Constants compiled by the University of Waterloo, decay constants tend to be in the magnitude of 2-10*ms*, though there are cases in which the decay constant is in the tens or hundreds of milliseconds [7]. The first study explores only a synaptic time constant value of 10*ms*, and the second study explores a range of values from 0*ms* to 50*ms*.

In contrast to both of the previous studies, this paper will aim to explore how introducing non-linear behaviour to the neuron models with varying degrees of biological realness and varying synaptic time constants will impact the performance of a pre-trained Keras-based Convolutional Neural Network utilizing a similar conversion method as the two previously-explored studies. In increasing order of biological realness, this paper will explore the Rectified Linear Spiking neuron model, the Leaky Integrate-and-Fire model, and the Adaptive Leaky Integrate-and-Fire model. These models will be tested across synaptic time constant values of 0.005s, 0.010s, 0.015s, and 0.020s. The CNN will be trained on the CIFAR-10 dataset – a sample of which is shown in Figure 1. This dataset is composed of 50,000 training images with 10 possible classifications and is publicly available at [8].

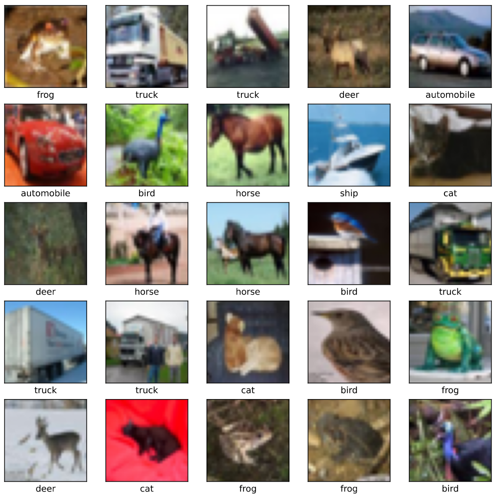

Figure 1: CIFAR-10 Sample Images

Methods

The model being explored is a purpose-built Convolutional Neural Network implemented in Keras that is composed of two pairs of successive convolutional and maximum pooling layers followed by another convolutional layer, a flattening layer, a fully-connected layer, and another fully-connected output layer. The Keras architecture is visualized in Figure 2 where the first Conv2D employs 32 filters and a 3x3 kernel, the first MaxPooling2D utilizes a 2x2 pool size, the second Conv2D employs 64 filters and a 3x3 kernel, the second MaxPooling2D utilizes a 2x2 pool size, the last Conv2D employs the same 64 filters and 3x3 kernel size as the previous, and the two remaining Dense layers are 64 neurons and 10 neurons respectively.

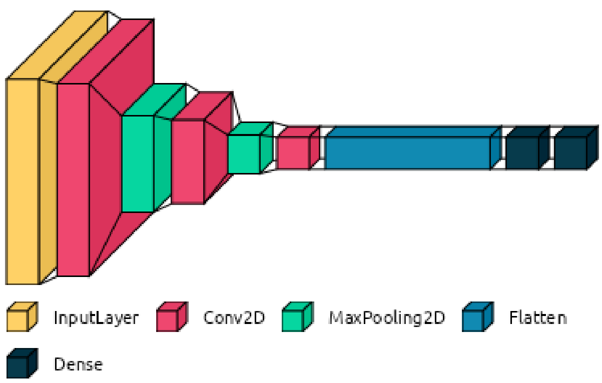

Figure 2: CNN Architecture

To introduce biological aspects into the model, the above CNN will be converted into an SNN using NengoDL [9]. Since training using NengoDL would be incredibly costly from a computation perspective, the Keras model was first pre-trained using the Adam optimization algorithm [10]. This Keras model was then evaluated on the CIFAR-10 testing set and showed a categorical accuracy of 71.03% and a categorical cross-entropy of 0.84. This provides one baseline to compare against, though elements of the NengoDL conversion may result in this being a poor baseline to compare to. Since the Keras-based model utilizes MaxPooling2D layers, it it important to note that these layers must be converted into average pooling layers in NengoDL, as selecting a "maximum" value in an SNN is a non-linear operation and cannot necessarily be implemented using spiking neurons with weighted connections [11]. This is further explored in the Discussion section. While the previously-explored studies built their Keras models using AvgPooling2D layers, this implementation is less typical of conventional CNNs, as maximum pooling operations are more able to perform local feature extraction [12]. This model modification necessitates a new baseline by which to compare the converted SNN: the use of Rectified Linear Rate (RLR) neurons. These neuron models perform in a very similar the same way to ReLU-activated neurons in the Keras model and will provide the most "ideal" performance of the model. Table 1 shows the summary of baseline metrics that can be used to compare the various SNNs to their original CNN and most idealized SNN (SNN-1). During evaluation, SNNs were run for 100 simulation timesteps with each timestep being 0.001*s* – or 0.1*s* total – and a maximum neuron firing rate (firing rate scaling) of 415*Hz*. Accuracy and Loss were calculated using Categorical Accuracy and Categorical Cross Entropy from the logit values of the last Dense layer (non-processed neuron values). This was primarily motivated by the lack of biological realness associate with Sigmoid and Softmax operations typically performed on outputs layers of multi-class classification problems. Due to the computation time of NengoDL simulations, performance evaluation was conducted on a sample of 1000 images within the testing set for all the Spiking Neural Networks.

Table 1: Baseline CNN and SNN Accuracy and Loss Performance

τ_{SYN}	***CNN***		***SNN-1 (RLR)***	
-	**Acc.**	**Loss**	**Acc.**	**Loss**
-	71.03%	0.84	-	-
0.005	-	-	39.32%	1.83
0.010	-	-	31.35%	2.03
0.015	-	-	24.06%	2.26
0.020	-	-	18.60%	2.43

Aside from the performance metrics above, we can also consider the output of the logits over time to observe how the SNN response changes with varying parameters. Figure 3 shows the logit response of SNN-1 to the first frog image in the training dataset (see Figure 4 in Appendix A). Evidently, the increase in synaptic time constant causes an increase in delay to value settling-time – thereby reducing the accuracy of the network at the 100th timestep.

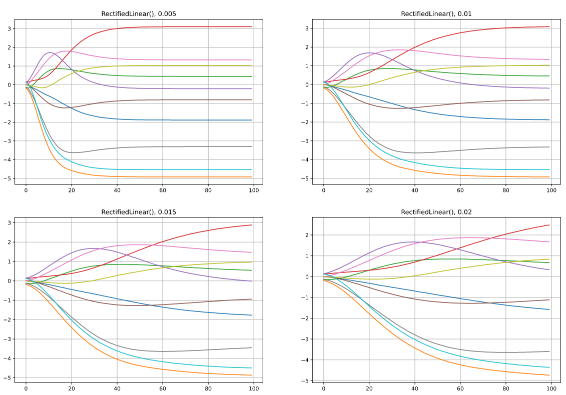

Figure 3: Logit Responses of SNN-1 for Varying Synaptic Time Constants (Top-Left: 0.005s, Top-Right: 0.010s, Bottom-Left: 0.015s, Bottom-Right: 0.020s)

SNN-2 was implemented using Spiking Rectified Linear neurons (SRL), SNN-3 was implemented using Leaky Integrate-and-Fire neurons (LIF), and SNN-4 was implemented using Adaptive Leaky Integrate-and-Fire (A-LIF). LIF and A-LIF have additional hyper-parameters to consider with respect to the membrane potential and spiking rates. LIF hyper-parameters include the membrane time constant, τ_{RC}, and absolute refractory period, τ_{REF}. While these hyper-parameters closely affect the spike train output, they will not be modified other than being set to $20ms$ and $2ms$ respectively. A-LIF hyper-parameters include the LIF hyper-parameters as well as an additional time decay factor and an adaptation state incrementer, τ_N and i_N respectively. These were set to 1 and 0.01 [3].

Results

The observed accuracies and loss values for the varying neuron models and synaptic constants are summarized in Table 2 alongside the baseline performance. Keras model accuracy and loss was additionally provided for reference. Of particular note is the ability of SNN-3 and SNN-4 to classify with higher synaptic time constants, as SNN-1 and SNN-2 both perform worse as the synaptic time constant increases. Additionally, the large decrease in accuracy between the Keras model and the best-performing SNN model is also of note. Logit responses of SNN-2, SNN-3, and SNN-4 with varying synaptic time constant values are included in Appendix A under Figure 5. Evidently, the logit responses of the non-linear neuron models indicate that they are more capable of settling to some classification value than the linear neuron models.

Table 2: Summary of Performance Metrics for Varying Spiking Neural Networks

τ_{SYN}	*CNN*		*SNN-1 (RLR)*		*SNN-2 (SRL)*		*SNN-3 (LIF)*		*SNN-4 (A-LIF)*	
-	**Acc.**	**Loss**	**Acc.**	**Loss**	**Acc.**	**Loss**	**Acc.**	**Loss**	**Acc.**	**Loss**
-	71.03%	0.84	-	-	-	-	-	-	-	-
0.005	-	-	39.32%	1.83	39.73%	1.73	38.21%	1.75	38.06%	1.75
0.010	-	-	31.35%	2.03	31.29%	2.01	37.19%	1.76	37.55%	1.76
0.015	-	-	24.06%	2.26	24.92%	2.19	36.75%	1.79	36.94%	1.78
0.020	-	-	18.60%	2.43	18.79%	2.41	35.73%	1.82	35.90%	1.82

Discussion

The results indicate that the non-linear neuron models are more robust against varying synaptic time constants. Despite this, all of the SNNs exhibit a decrease in performance as the synaptic time constant increases. Rectified Linear neurons – RLR and SRL – both see an approximately 21% decrease in accuracy between minimum and maximum synaptic time constant values, while non-linear neurons – LIF and A-LIF – both see an approximately 3% decrease in accuracy. Interestingly, the more biologically-inspired A-LIF model results in an SNN that is more accurate than the LIF model SNN for all but one synaptic time constant value (0.005*s*). The Rectified Linear SNNs also exhibit higher accuracy than the non-linear SNNs for the minimum synaptic time constant value.

The large decrease in classification accuracy between the Keras-based CNN and the Nengo-based SNNs is fairly different from the previously explored literature, though this could likely be attributed to the transition from the maximum pooling layers to average pooling layers. From a neurological perspective, this change is necessitated by the relationship between simple and complex cells in the primary visual cortex (V1). Since complex cells are connected to a pooling of simple cells, they perform some pooling of the simple cells' receptive fields to extract features [3]. The previous studies built their CNNs using average pooling layers, lending to their convertibility to SNNs, but it is less typical of modern CNNs to use average pooling layers over maximum pooling layers.

Despite the loss in accuracy due to maximum and average pooling conversions, it is particularly interesting that SNN-3 and SNN-4 perform nearly as well as minimum synaptic time constant configurations of SNN-1 and SNN-2. Since SNN-1 is effectively the baseline of SNN performance, the results indicate that non-linear neuron models can perform nearly as well as (if not better than) linear neuron models with no additional tuning for varying time parameters.

In contrast to the logit curves observed in SNN-1, the logit curves observed in SNN-2, SNN-3, and SNN-3 are significantly less smooth, though increases in the synaptic time constant increase the response smoothness. Less desirable is the consequent increase in settling time for the SNNs, lending to worse accuracy for the limited amount of simulation time.

Conclusion

The above results and following discussion indicate that Spiking Neural Networks provide a viable alternative to conventional Artificial Neural Networks for implementations of Convolutional Neural Networks performing image classification tasks. With 1.1% and 1.3% decreases in accuracy from the Rectified Linear Rate neuron model, both the Leaky Integrate-and-Fire and the Adaptive Leaky Integrate-and-Fire Spiking Neural Networks provide viable and more temporally robust classification than their alternatives. Future work could be directed by the re-training or tuning of the converted model to some sample of the training set. Additionally, the dense layer before the output layer could be made larger such that the converted network is more capable of representing and reaching some constant values. This may necessitate some re-exploration of the original Keras model and its number of feature maps per convolutional and pooling layers. Lastly, one could explore the impact of using an average pooling layer during Keras training rather than the maximum pooling layer currently being used.

All code for the project can be found at the associated GitHub repository under the "Project" section [13].

References

[1] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, no. 4, pp. 193–202, Apr. 1980, doi: https://doi.org/10.1007/bf00344251

[2] R. De Man, G. J. Gang, X. Li, and G. Wang, "Comparison of deep learning and human observer performance for detection and characterization of simulated lesions," *Journal of Medical Imaging*, vol. 6, no. 02, p. 1, Jun. 2019, doi: https://doi.org/10.1117/1.jmi.6.2.025503

[3] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. A. Siegelbaum, and A. J. Hudspeth, *Principles of Neural Science*, 5th ed. New York ; Toronto: Mcgraw-Hill Medical, 2013.

[4] V. Honzik and R. Moucek, "Spiking Neural Networks for Classification of Brain-Computer Interface and Image Data," *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Dec. 2021, doi: https://doi.org/10.1109/bibm52615.2021.9669864

[5] B. Wang *et al.*, "Firing Frequency Maxima of Fast-Spiking Neurons in Human, Monkey, and Mouse Neocortex," *Frontiers in Cellular Neuroscience*, vol. 10, Oct. 2016, doi: https://doi.org/10.3389/fncel.2016.00239

[6] Thu Quyen Nguyen, Quoc Trung Pham, Phuong Chi Hoang, Quang Hieu Dang, Duc Minh Nguyen, and Huy Hoang Nguyen, "An improved spiking network conversion for image classification," Oct. 2021, doi: https://doi.org/10.1109/mapr53640.2021.9585199

[7] CNRG UWaterloo, " Neurotransmitter Time Constants (PSCs)," *compneuro.uwaterloo.ca*. https://compneuro.uwaterloo.ca/research/constants-constraints/neurotransmitter-time-constants-pscs.html

[8] A. Krizhevsky, "CIFAR-10 and CIFAR-100 datasets," *Toronto.edu*, 2009. https://www.cs.toronto.edu/~kriz/cifar.html

[9] "Deep learning integration for Nengo — NengoDL 3.6.1.dev0 docs," *www.nengo.ai*. https://www.nengo.ai/nengo-dl/introduction.html

[10] "Keras documentation: Adam," *keras.io*. https://keras.io/api/optimizers/adam/

[11] "Nengo Loihi: Computable functions and Spiking Behaviour," *Nengo forum*, Mar. 17, 2021. https://forum.nengo.ai/t/nengo-loihi-computable-functions-and-spiking-behaviour/1646/4

[12] L. Zhao and Z. Zhang, "A improved pooling method for convolutional neural networks," *Scientific Reports*, vol. 14, no. 1, p. 1589, Jan. 2024, doi: https://doi.org/10.1038/s41598-024-51258-6

[13] JakebC, "borealis31/SYDE552," *GitHub*, Apr. 23, 2024. https://github.com/borealis31/SYDE552

Appendix A

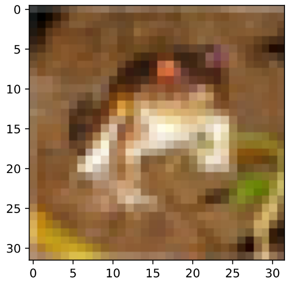

Figure 4: Frog Image #1 for Logit Responses

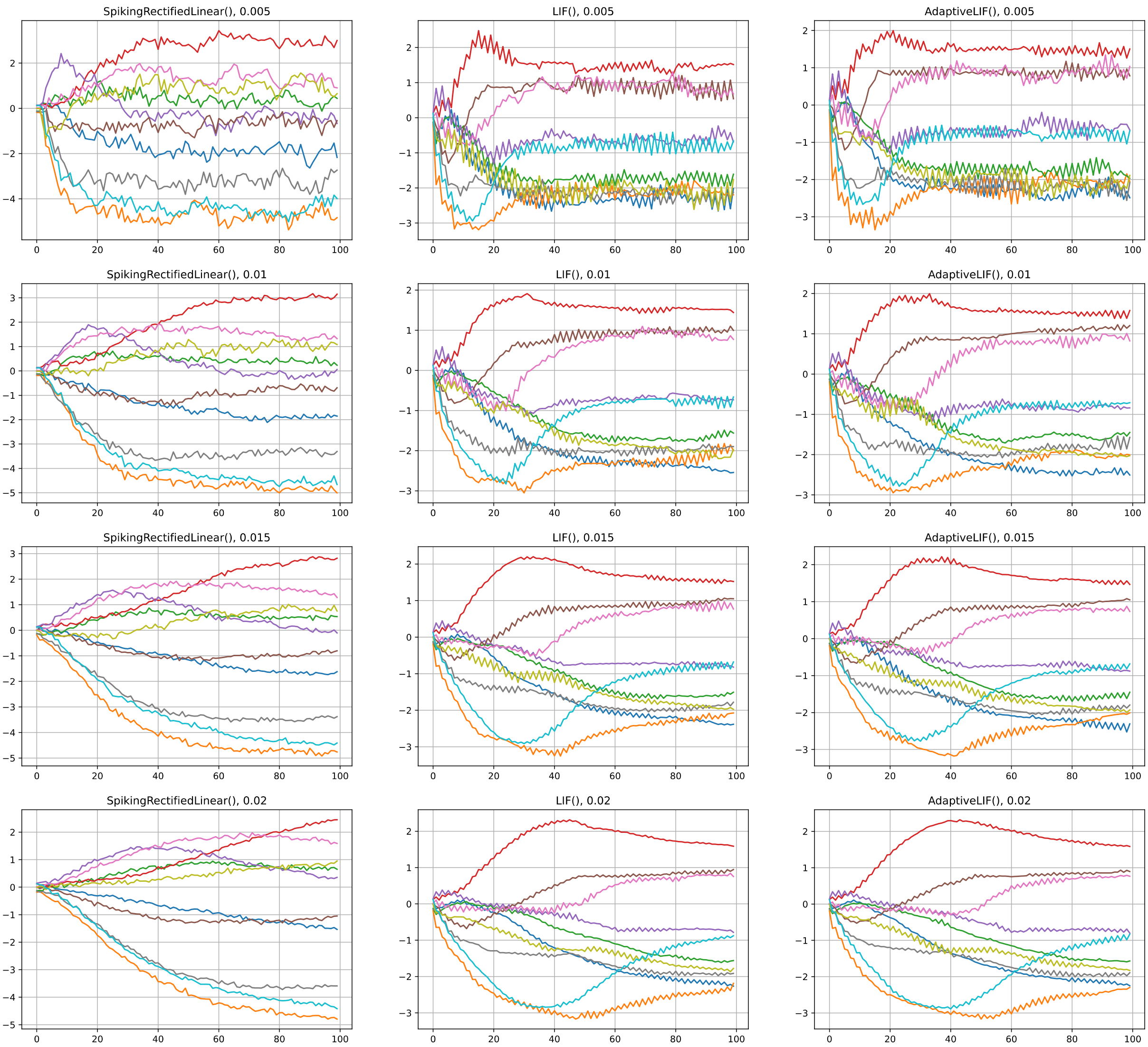

Figure 5: Logit Responses to Frog Image #1 for Varying Neuron Models and Synaptic Time Constants