Conversion of ConvNets to Spiking Neural Networks With Less Than One Spike per Neuron

Javier López-Randulfe (lopez.randulfe@tum.de)[†] **Nico Reeb (nico.reeb@tum.de)**[†]

Alois Knoll (knoll@in.tum.de)[†]

[†]Department of Informatics
Technical University of Munich
Munich, Germany

Abstract

Spiking neural networks can leverage the high efficiency of temporal coding by converting architectures that were previously learnt with the backpropagation algorithm. In this work, we present the application of a time-coded neuron model for the conversion of classic artificial neural networks that reduces the computational complexity in the synaptic connections. By adapting the ReLU activation function, the network achieved a sparsity of 0.142 spikes per neuron. The classification of handwritten digits from the MNIST dataset show that the neuron model is able to convert convolutional neural networks with several hidden layers.

Keywords: spiking neural network; time coding; deep learning; object classification; spiking network conversion

Introduction

The main mechanism for transmitting information in the brain are spikes, by either using specific spike times or spike rates over a time period. Research has shown that the reaction time of the brain for certain tasks like face detection is very low, allowing only few spikes per neural layer involved (Martin, Davis, Riesenhuber, & Thorpe, 2018). Moreover, time coding is more efficient, which could also explain the low power consumed by the brain. Multiple studies have also shown that only few neurons spike at similar times in the brain, pointing towards a high sparsity in the neural activity.

Many algorithms developed within computer science have brought the performance of deep neural networks closer to biology over the last decades. The high synaptic density of dense neural layers has been dramatically reduced by using convolutional kernels that take advantage of the translation invariance present in natural data. Furthermore, the well-known Rectified Linear Unit (ReLU) activation function provides a high degree of sparsity by maintaining neurons inactive unless their aggregated input is positive. As feature maps in convolutional layers respond to specific input patterns, most of the neurons stay silent for a given sample. Recently, research has also focused on replacing the floating point arithmetic of artificial neural networks (ANNs) by spikes with the aim of increasing their efficiency. In the case of supervised tasks, the strategy that provides the best performance is based on training a traditional ANN using the backpropagation algorithm, and then converting the ANN to a spiking neural network (SNN). Even though initial attempts of converting ANNs to SNNs used spike rates (Rueckauer, Lungu, Hu, Pfeiffer, & Liu, 2017; Blouw & Eliasmith, 2020), the focus is shifting to the conversion into sparse time-coded spike trains (Stöckl & Maass, 2021). On current hardware only SNNs with an expected number of spikes < 1.72 per output can consume less energy than comparable ANNs (Davidson & Furber, 2021).

In this work, we adapt the neuron model developed in (Lopez-Randulfe et al., 2022) for converting ANNs into SNNs using no more than one spike per neuron. We tested the model on a multi-layer convolutional neural network (CNN) on the handwritten digit dataset MNIST.

Neuron and network model

The neuron model relies on linear time-to-first-spike encoding, and it takes data x that is normalized to the range $[0, x_{\text{max}}]$ due to the ReLU activation function. Therefore, we simplify the encoding from (Lopez-Randulfe et al., 2022) to

$$t(x) = \frac{t_{\text{max}}}{x_{\text{max}}} \cdot (x_{\text{max}} - x)\,, \tag{1}$$

which maps a real value x to a single spike in the time domain $t \in [0, t_{\text{max}}]$.

Two-stages neuron model

The neuron model performs the multiplication $u = W \cdot x$ between an NxM matrix W and an M dimensional vector x over two stages in the time domain, i.e., *silent* and *spiking* stage, respectively.

During the *silent stage*, neuron i processes the incoming spikes from causal neurons $\Gamma_i^<$ following the voltage dynamics

$$u_i(t) = \sum_{j \in \Gamma_i^<} W_{ij}(t - t_j)\,, \tag{2}$$

with t_j being the spike time of input neuron j, and W_{ij} the element of the weight matrix connecting neuron i and j.

If we use (1) for encoding the input x, the voltage (2) after the *silent* stage gives the scaled result of the scalar product

$$u_i(t_{\text{max}}) = \frac{t_{\text{max}}}{x_{\text{max}}} \sum_{j=0}^{M-1} W_{ij} \cdot x_j\,. \tag{3}$$

During the *spiking stage*, the neuron maps the voltage $u_i(t_{\text{max}})$ to spike times by charging it with a constant current I_{ext} and producing a spike when u_i reaches a given threshold

u_{th}. The output spike time follows the same encoding as in (1).

Tuning the model for converting deep nets

The tuning of the neuron model involves setting the values of the constants u_{th}, I_{ext}, and t_{max}.

For representing a ReLU function, we need to map the positive voltages in (3), i.e., $u \in (u_{\text{min}} = 0, u_{\text{max}}]$ to spike times. Thus, negative and zero values are represented by the absence of a spike, granting the resulting SNN a high degree of sparsity. We obtain this behaviour by setting I_{ext} to

$$I_{\text{ext}} = \frac{u_{th} - u_{\text{min}}}{t_{\text{max}}} = \frac{u_{th}}{t_{\text{max}}}\,. \tag{4}$$

Moreover, for mapping the whole range of u, u_{th} needs to be equal to the largest possible voltage u_{max}

$$u_{\text{max}} = max\{u(t_{\text{max}})\} = \sum_j W^+ t_{\text{max}}\,, \tag{5}$$

where W^+ is the subset of positive weights. In general, setting $u_{th} = u_{\text{max}}$ leads to a poor resolution, as natural data typically use a portion of the whole range. Alternatively, we determined empirically a lower u_{th} that minimizes the error of the conversion for the used data depending on t_{max} (see Figure 2).

Finally, t_{max} is fixed as a trade-off between the SNN precision and its computational complexity.

From this description, the proposed model of (Lopez-Randulfe et al., 2022) can be used for generic matrix-vector multiplications. Since the spike outputs of the neurons utilize the same encoding as the input spikes, a naturally stacking of populations is possible and allows the replication of ANNs.

Experiment results

We have tested the neuron model by converting a previously trained convolutional neural network (CNN) for the classification of handwritten digits from the MNIST dataset. The trained CNN consists of two convolutional layers, two max pooling layers (one after each convolutional layer), and two dense layers (see Figure 1). For training, we used a dropout of 0.2 before each of the dense layers, and we augmented the training set by rotating, scaling, and shifting the original images.

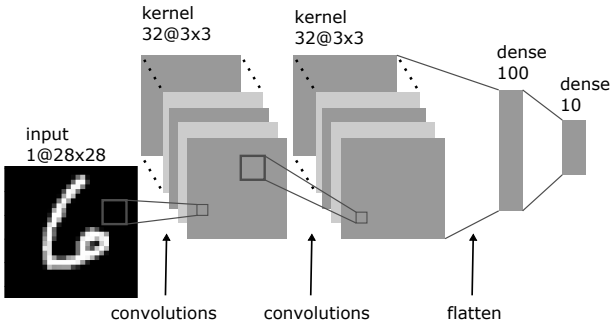

Figure 1: CNN architecture for MNIST data set. Each convolutional and dense layer uses a ReLU activation function.

Table 1 shows the performance summary of the CNN and the converted SNN. We recorded the number of floating point operations (FLOPs) required for the synaptic connections between the different layers of the CNN, and the number of spike operations for the equivalent SNN. Due to the nature of the converted ReLU, the spiking neurons with a negative internal state do not produce a spike after integrating the inputs, whereas neurons with a positive state produce a single spike. This results in the SNN emitting on average 0.142 spikes per neuron.

Table 1: Performance comparison a traditional CNN and its equivalent converted SNN on the MNIST dataset.

	FLOPs	synaptic ops.	Acc.
ANN	14316496	-	99.56
SNN	-	1015015	99.44

We have also analyzed the impact of the total simulation steps and the threshold voltage, which are the two hyperparameters that the spiking neuron model introduces with regard to the original ANN. Figure 2 depicts the impact of these two parameters on the root-mean-squared error between the output of the SNN and the ANN for a single convolutional layer.

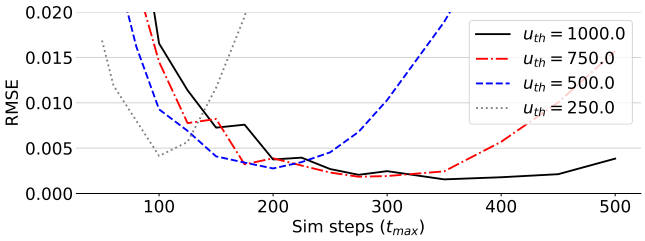

Figure 2: Error between the output of a convolutional layer and its spiking equivalent.

Conclusion

In this paper, we have tested the performance of a time-coded neuron model for converting convolutional neural networks into SNNs. The results of the experiments show a considerable reduction in the number of synaptic operations required during inference, while maintaining a small loss in accuracy.

Further experiments shall assess the performance of the model for more complicated architectures and datasets, as well as providing a comparison between the energy consumed by the SNN and the original ANN. Future research can also focus on the encoding of the information for increasing the sparsity of the SNN. Coding techniques like rank or M-of-N encoding have already shown high performance while achieving a big reduction in the amount of required spikes. Moreover, the model could incorporate prior information about the input data for fine tuning parameters like the encoding range or the voltage threshold of the neurons.

References

Blouw, P., & Eliasmith, C. (2020). Event-driven signal processing with neuromorphic computing systems. In *Icassp 2020-2020 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 8534–8538).

Davidson, S., & Furber, S. B. (2021). Comparison of artificial and spiking neural networks on digital hardware. *Frontiers in Neuroscience*, *15*. Retrieved from https://www.frontiersin.org/article/10.3389/fnins.2021.651141 doi: 10.3389/fnins.2021.651141

Lopez-Randulfe, J., Reeb, N., Karimi, N., Liu, C., Gonzalez, H., Dietrich, R., ... Knoll, A. (2022). Time-coded spiking fourier transform in neuromorphic hardware. *IEEE Transactions on Computers*, 1-1. doi: 10.1109/TC.2022.3162708

Martin, J. G., Davis, C. E., Riesenhuber, M., & Thorpe, S. J. (2018). Zapping 500 faces in less than 100 seconds: evidence for extremely fast and sustained continuous visual search. *Scientific reports*, *8*(1), 1–12.

Rueckauer, B., Lungu, I.-A., Hu, Y., Pfeiffer, M., & Liu, S.-C. (2017). Conversion of continuous-valued deep networks to efficient event-driven networks for image classification. *Frontiers in neuroscience*, *11*, 682.

Stöckl, C., & Maass, W. (2021). Optimized spiking neurons can classify images with high accuracy through temporal coding with two spikes. *Nature Machine Intelligence*, *3*(3), 230–238.