In Plane Sight: Scanlines for Spiking Neural Networks

Joseph Mills
University of Sussex
jm662@sussex.ac.uk

Abstract

Scan Line Encoders aim to reflect the movement of the human eye in processing training data. This work explores the use of these encoders on the effective training of Spiking Neural Networks (SNN) on the MNIST dataset. The training of these networks uses the EventProp, event-based, training algorithm to compute the exact gradient. It is shown that scanlines are a very promising area for future research across computational science, neuroscience and neuromorphic hardware.

1 Introduction

Learning mechanisms are limited not only by what information they are presented but also by how they are presented. Recently, great focus has been on neuron spiking behaviour, learning theory and data encoding in the development of neural networks that are biologically plausible, computationally efficient, and also amenable to being used within neuromorphic hardware [3, 7, 9, 10].

Many other areas of machine learning have focused on embodiment, the act of giving machine learning systems similar sensory inputs to that of biological systems, with much of this focus having centred around AI in robotics. This work has used more biologically inspired inputs into conventional machine learning frameworks.

A component of emulating the movement of human eyes is known as Saccadic Eye Movement (SEM) and can be thought of as the discrete linear movements of the eye. This is what Scan Line Encoders aim to replicate. Mapping of saccade dates back to 1998 [1] and since then, research shows how changes to the movements could also be early indicator of neuro-degenerative disease [8].

This paper builds upon the work on intel's Loihi and Gardner's first-to-spike decoding work where Scan Line Encoders have been used [4, 6] on Spiking Neural Networks (SNN). It is also the first implementation of Scan Line Encoders being used with the EventProp [5] algorithm, in the literature reviewed.

2 Methods

To effectively evaluate Scan Line Encoders, it was important to present a multitude of scanline encoding techniques, and to lay foundation for future work to develop these ideas further.

Akin to many computer vision classification tasks the standard MNIST database consiting of $60,000$ training images and $10,000$ test images [2]. No additional pre-processing or normalisation took place.

These input images are passed through two separate Scan Line Encoders with various input parameters and are presented to the SNN. The trained models are then tested on the MNIST test set and the sparse categorical accuracy was output. As seen in Table 2

All work was completed on Google Colab on one A100 GPU with 83GB of RAM and 40GB of GRAM and utlising mlGeNN, the open source library optimised for SNN's.

MRes Advanced Artificial Intelligence (University of Sussex 2024).

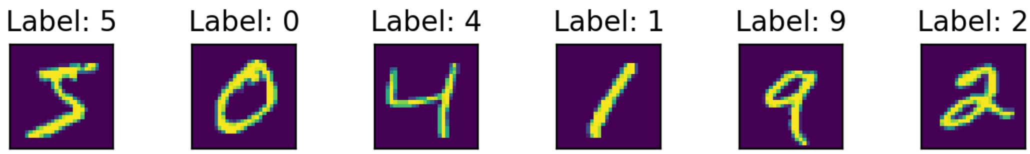

Figure 1: Example of Standard MNIST

2.1 Scan Line Encoding

2.1.1 Scan Line Generator

Initially, scanlines were generated to match the original work on Loihi [4] and by means of understanding the underlying processes. Thereafter, and used throughout, scanlines were randomly generated.

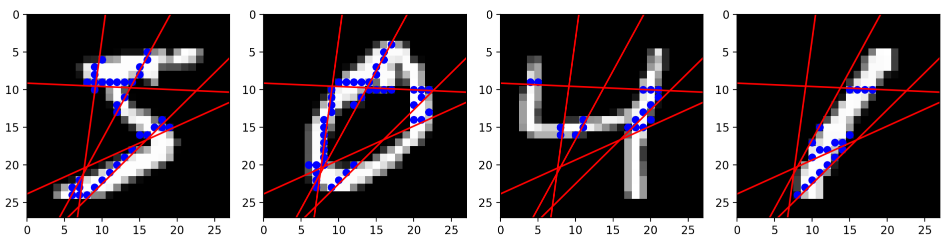

Figure 2: Example of scanlines (red) and the fired pixels (blue) using 5 scanlines

A pair of x and y co-ordinates were generated from the uniform distribution in the range (7,21), as to reflect the middle box within the input image with dimension 28x28. The two co-ordinates were solved to find the gradient m and intercept c. The scanlines were fixed through training as seen in Figure 2 and, for example, the scanlines used for ***sl-A*** with 10 lines are the same lines as **sl-B** with 10 lines, as depicted in Table 2.

2.1.2 Encoding Strategies

Two approaches were taken to explore the use of scanlines.

First, a simple encoder iteratively processed linearly, all pixels that intersect with the scanline. This is different to that described in Loihi et al. and Gardner et al. [4, 6]. However, this is useful given that the dimensionality of the model, namely the input spiking neurons, matches that of the log latency encoder. This encoder is referred to as ***sl-A***. In Figure 2, for ***sl-A***, all fired pixels (blue) were passed to the model.

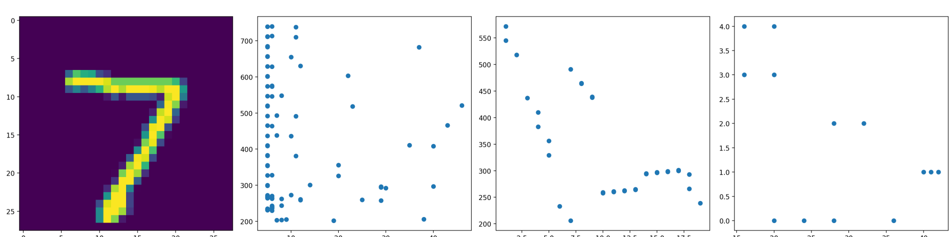

Figure 3: MNIST (left), latency encoding (centre-left), ***sl-A*** encoding (center-right), ***sl-B*** encoding (right) for 5 random sequences. Where the x-axis is the spike time and the y axis is the input neuron id.

The second approach closely followed that described by Gardner et al. [6]. For this, the number of input spiking neurons equalled the number of scanlines that were used. Instead of from bottom-to-top with 9 discrete time steps, like Gardner et al. [6], the image is processed top-to-bottom in 12, discrete time steps. If a scanline intersects with a $pixelvalue > 0$ within that time step, the neuron associated with the scan-line fires. The times were then multiplied by a scaling factor of 5. The scaling factor was quickly identified as very important in the effective training of the model. From the range 0 - 5, 5 appeared to have the best results at lower epochs. Equally, the membrane refractory constant, τ_{refrac} could have also been optimised instead. This encoder is named ***sl-B***.

In Figure 3 the input spikes for an example MNIST image are shown for ***sl-A*** and ***sl-B***. It is clear in **sl-b** how sparse the input spikes are.

2.2 EventProp

Unique to this paper is the evaluation of Scan Line Encoders using the EventProp algorithm [5].

Given the loss functions $l_V(V,t)$, $l_\mathrm{p}(t_{post})$ that depend on the membrane potentials V, time t and the set of post-synaptic spike times t_{post}. The total loss for EventProp is given by

$$\mathcal{L} = l_p(t_{post}) + \int_0^T l_V(V(t),t)dt \tag{1}$$

The loss derivative with respect to a specific weight $w_{ji} = (W)_{ji}$ that connects pre-synaptic neuron i (the firing neuron) to post-synaptic neuron j (the receiving neuron) is the sum over the spikes caused by i,

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}w_{ji}} = -\tau_{\text{syn}} \sum_{\text{spikes from } i} (\lambda_I)_j, \tag{2}$$

λ_I is the adjoint variable (Lagrange multiplier) corresponding to the synaptic current I. Equation 2 therefore samples the post-synaptic neuron's adjoint variable $(\lambda_I)_j$ at the spike times caused by neuron i.

2.3 Model Structure

Each model was build with 3 layers consisting of input neurons of size N_{Input}, hidden layer consisting of Leaky Integrate and Fire (LIF) neurons with size N_{Hidden} and the output layer, matching the MNIST classes, of size N_{Output}. The models were trained with various different input neurons, as in Table 2. All other parameters remained the same.

Parameter	***Value***
N_{Input}	5,10,15,20,784
N_{Hidden}	128
N_{Output}	10
V_{Thresh}	0.61
τ_{mem}	20ms
τ_{refrac}	5ms
$optimiser$	adam
$epochs$	5,100

Table 1: Parameters of the trained models.

Where V_{Thresh} is the membrane voltage firing threshold, τ_{mem} is the membrane voltage time constant and τ_{refrac} is the duration of the refractory period.

2.4 Log Latency Encoding

To act as a baseline for the experiments and for what is used in many SNN papers. Log Latency Encoding was used.

A benefit of Log Latency Encoding is the sparsity of spikes. It is assured that a particular neuron only fires once in the time the image is presented to the model.

Log Latency Encoding involves encoding the spike times as a function of the pixel values, x that exceed a particular threshold θ, multiplied by a scaling factor, τ_{eff}.

$$T(x) = \begin{cases} \tau_{\text{eff}} \log\left(\frac{x}{x-\theta}\right) & x > \theta \\ \infty & otherwise \end{cases} \tag{3}$$

Here, $\tau_{\text{eff}} = 20\text{ms}$ and $\theta = 51$ were used.

3 Results

	Parameters		**SCA ↑**	
	Input Neurons	Scanlines	Epochs	
Encoder	N_{Input}	N	5	100
log latency	784		0.93	0.97
sl-A	784	5	0.83	0.83
sl-A	784	10	0.89	0.92
sl-A	784	15	0.89	0.94
sl-A	784	20	**0.90**	**0.94**
sl-B	5	5	0.52	0.51
sl-B	10	10	0.65	0.66
sl-B	15	15	**0.75**	0.74
sl-B	20	20	0.74	**0.75**

Table 2: Results showing model parameters and Sparse Categorical Accuracy (SCA) on MNIST Test

4 Discussion

The two encoding strategies proved to be very strong alternatives to log latency encoding, particularly when intending to reduce the dimensionality of the model as in ***sl-B***. Given the low number of training epochs completed it is of no surprise that the performance of ***sl-B*** falls short of the 1600 epochs used by Gardner et al. [6].

In future work, optimising the firing times in ***sl-B*** would likely hold most value. It can be seen that there is negligible different between the two different epochs for **sl-B** which is not what we would expect given the sparseness of spikes. This would require further investigation and experimentation with different hyper-parameters.

What would also be of great interest is a combination of ***sl-A*** and ***sl-B*** encoding strategies. Whereby 784 input neurons are associated with the pixel grid, for MNIST, and N neurons are associated with the id of the associated scanline. This pairing would likely emulate the internal view of the images and the underlying physiological process that fires said neurons.

In the associated code, an additional scanline generator was created to emulate the fovea. This foveated scanline generate is compatible with ***sl-A*** and ***sl-b*** and input's spikes if there are $pixelvalues$ > 0 in the adjecent pixels. This tries to capture the field of focus the eye exhibits. This, too, showed very promising results.

Another fault of this research was both not repeating experiments due to the random initialisation of weights or repeating the experiments with different randomly generated scanlines. This would have ensured that the results are not reflective of the specific scanlines used or the initialisation weights but instead more reflective of the strategy of Scan Line Encoding. Given that different scanlines were used each experiment with varying length of random scanlines, as in 2, it can be assumed this had little effect. However, in future work, this would be of great value.

5 Conclusion

Log latency encoders still comes out on-top for a smaller number of epochs, in the results of this paper. Despite this, scanlines are seen to be a promising area of future research, with numerous ways in which to encode scanlines and optimise systems for their use. Though training iterations may need to increase with the sparseness of spikes, as seen in Figure 3 the saving in potential inference time could be staggering. From neuroscience to computational science, scanlines could open a door to the intrinsic link between neuronal firing behaviour observed in biology and the physiological processes required to effectively utilise that behaviour.

References

[1] Itti, Laurent / Koch, Christof / Niebur, Ernst(1998): *A model of saliency-based visual attention for rapid scene analysis*.

[2] Deng, Li(2012): *The MNIST Database of Handwritten Digit Images for Machine Learning Research*.

[3] Kerr, Dermot / Coleman, Sonya / McGinnity, TM / Wu, Qingxiang / Clogenson, Marine(2012): *A novel approach to robot vision using a hexagonal grid and spiking neural networks*.

[4] Lin, Chit Kwan / Wild, Andreas / Chinya, Gautham / Cao, Yongqiang / Davies, Mike E. / Lavery, Daniel M. / Wang, Hong(2018): *Programming Spiking Neural Networks on Intel's Loihi*.

[5] Wunderlich, Timo C. / Pehle, Christian(2021): *Event-based backpropagation can compute exact gradients for spiking neural networks*.

[6] Gardner, Brian / Grüning, André(2021): *Supervised Learning with First-to-Spike Decoding in Multilayer Spiking Neural Networks.*

[7] Auge, Daniel / Hille, Julian / Mueller, Etienne / Knoll, Alois(2021): *A Survey of Encoding Techniques for Signal Processing in Spiking Neural Networks*.

[8] Boz, Hatice Eraslan / Koçoğlu, Koray / Akkoyun, Müge / Tüfekci, Işıl Yağmur / Ekin, Merve / Akdal, Gülden(2023): *Eye movement patterns during viewing face images with neutral expressions in patients with early-stage Alzheimer's disease and amnestic mild cognitive impairment*.

[9] Barchid, Sami / Allaert, B. / Aissaoui, Amel / Mennesson, José / Dj'eraba, Chaabane(2023): *Spiking-Fer: Spiking Neural Network for Facial Expression Recognition With Event Cameras*.

[10] D'Angelo, Giulia / Bartolozzi, Chiara(2024): *OP-10Bioinspired intelligent visual attention system for the humanoid robot iCub exploring event-driven sensing and neuromorphic hardware*.