Training Probabilistic Spiking Neuron Networks with First-to-spike Decoding

Thariq Shanavas(1) Ravi Kumar Kushawaha(1)

EE-746 Neuromorphic Engineering

Overview

- Neurons in the human brain communicate by means of sparse spiking processes. As a result, they are mostly inactive, and energy is consumed sporadically. Third-generation neural networks, or Spiking Neural Networks (SNNs), aim at harnessing the energy efficiency of spike-domain processing by building on computing elements that operate on, and exchange, spikes.
- Proof-of-concept implementations have shown remarkable energy savings by multiple orders of magnitude with respect to second-generation neural networks. Probabilistic models have the capability of learning firing thresholds using standard gradient based methods, while in deterministic models these are instead treated as hyper-parameters.

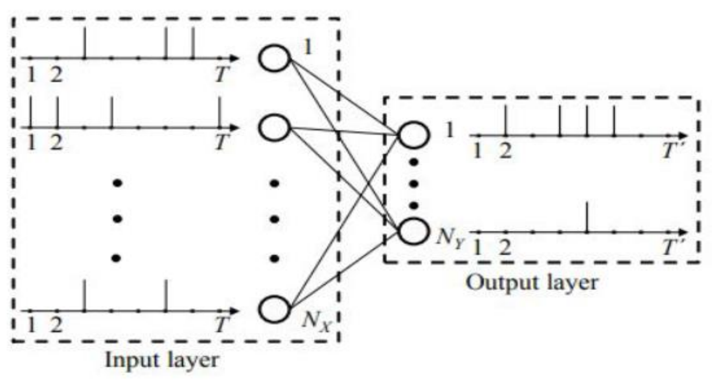


Fig. 1. Two-layer SNN for supervised learning.

- We study here a first-to-spike decoding rule, whereby the SNN can perform an early classification decision once a spike firing is detected at an output neuron. This generally reduces decision latency and complexity during the inference phase.
- We propose the use of flexible and computationally tractable Generalized Linear Model (GLM). We then derive a novel SGD-based learning algorithm that maximizes the likelihood that the first spike is observed at the correct output neuron.

Theoretical Model

- The SNN is fully connected and has *Nx* presynaptic neurons in the input,or sensory layer, and *Ny* neurons in the output layer. Each output neuron is associated with a class.
- With the conventional **rate encoding** method, each entry of the input signal is converted into a discrete-time spike train by generating an independent and identically distributed (i.i.d.) Bernoulli vectors.
- The probability of generating a "1", i.e., a spike, is proportional to the value of the energy.
- The relationship between the input spike trains from the NX presynaptic neurons and the output spike train of any postsynaptic neuron I follows a **GLM neuron model.**

$$u_{i,t} = \sum_{j=1}^{N_X} \alpha_{j,i}^T \mathbf{x}_{j,t-\tau_y}^{t-1} + \beta_i^T \mathbf{y}_{i,t-\tau'_y}^{t-1} + \gamma_i,$$

- Ui,t = membrane potential of an output neuron i at time t
- αj;i is a vector that defines the synaptic kernel(SK) applied on the {j; i} synapse between presynaptic neuron j and postsynaptic neuron i
- βi is the feedback kernel (FK) and γi is a bias parameter.

- The Synaptic Kernel and Feedback Kernel filters are parameterized as the sum of fixed basis functions with learnable weights.
- Implemented using Raised Cosine basis vectors.

Training with First to Spike decoding

- During the inference phase, with first-to-spike decoding, a decision is made once a first spike is observed at an output neuron. It follows naturally that the objective function to be maximized would be the probability that the target neuron fires first..
- Let the probability of the output neuron i firing at time sample t be , where g is the sigmoid function.
- Probability that target neuron c fires first at time t,

$$p(t) = \left[\Pi_{t'=1}^{t-1}\Pi_{i=1}^{N_y}\{1-g(u_i(t'))\}\right] * \left[\Pi_{i=1,i\neq c}^{N_y}\{1-g(u_i(t))\}\right] * g(u_c(t))$$

- The cost function to be minimized is the negative log of , summed over all time samples per simulation, T.

$$L = -\log\left(\Sigma_{t=1}^{T} p(t)\right)$$

We performed gradient descent to minimize the cost function.

Figures

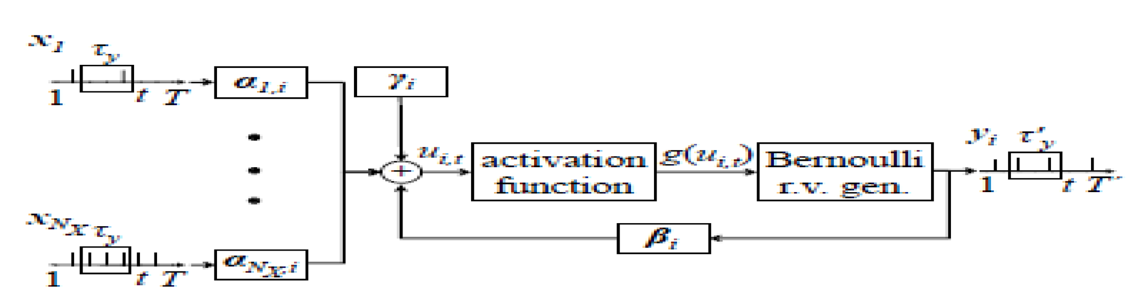


Fig. 2. GLM neuron model.

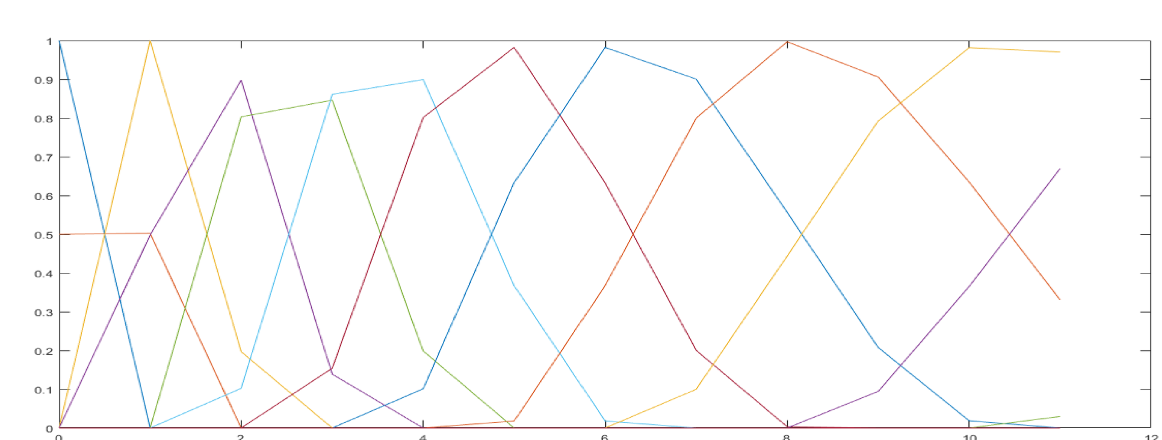


Fig. 3. Raised Cosine Basis vectors

- We have used a simplified dataset with two target classes, with four input neurons. Analogous to the MNIST encoding in the paper, the two classes correspond to input neuron spiking probabilities of (0.5,0.5,0.05,0.05) and (0.05,0.05,0.5,0.5).
- The training is done on a dataset of 200 samples, while the accuracy is tested on a different dataset of 100 samples.

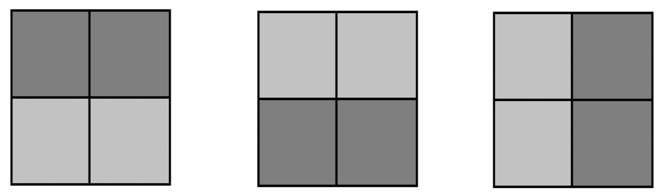
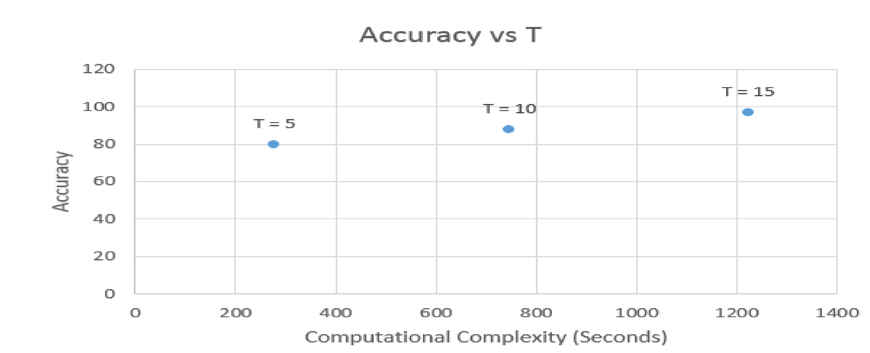


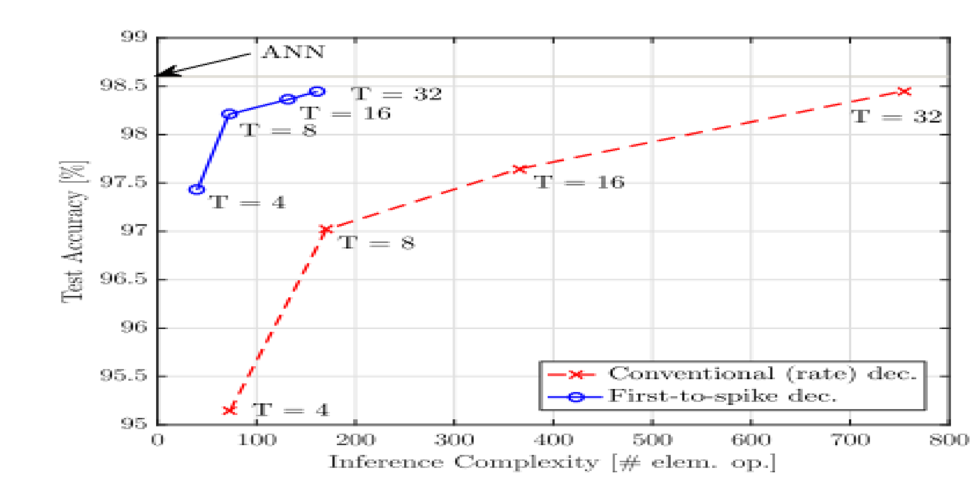
Fig. 4. three classes of images

Results

- The accuracy as a function of T is presented below.



- Comparing with results in the paper may not make much sense now, since we operated on different datasets. Following are the results from the paper we implemented



- Added 5% Gaussian noise – analogous to handwriting variations in MNIST dataset. Using MNIST dataset is prohibitively expensive.

Conclusion

- Since the neuron model is probabilistic, the spike pattern may not reflect the correct input figure – for instance, for class 1, the four input neurons spike with probability 0.5,0.5,0.05,0.05. The spike pattern may easily be misinterpreted. In comparison, there are 400 input neurons for MNIST.
- The SNN is otherwise demonstrated to be able to perform classification.

(1) Indian Institute of Technology Bombay