TRAINING PROBABILISTIC SPIKING NEURAL NETWORKS WITH FIRST-TO-SPIKE DECODING

Thariq Shanavas, and Ravi Kumar Kushawaha
Department of Electrical Engineering, Indian Institute of Technology Bombay, Mumbai, India.
Email: thariqshanavas@iitb.ac.in
Department of Electrical Engineering, Indian Institute of Technology Bombay, Mumbai, India.
Email: 150070045@iitb.com

1. INTRODUCTION

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. Most existing algorithms are based on variations of the unsupervised mechanism of Spike-Timing Dependent Plasticity (STDP), which updates synaptic weights based on local input and output spikes. In the context of SNNs, probabilistic models have the capability of learning firing thresholds using standard gradient based methods, while in deterministic models these are instead treated as hyperparameters and set by using heuristic mechanisms such as homeostasis. In this paper, we study the problem of training the two-layer SNN illustrated in Fig. 1 under a probabilistic neuron model, for the purpose of classification. 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.

2. SPIKING NEURAL NETWORK WITH GLM NEURONS

In this section, we describe the architecture of the two-layer SNN under study and then we present the proposed GLM neuron model. We consider the problem of classification using a two-layer SNN. As shown in Fig. 1, the SNN is fully connected and has N_X presynaptic neurons in the input, or sensory layer, and N_Y neurons in the output layer. Each output neuron is associated with a class. In order to feed the SNN, an input example, e.g., a gray scale image, is converted to a set of N_X discrete-time spike trains, each with T samples, through rate encoding. The input spike trains are fed to the N_Y postsynaptic GLM neurons, which output discrete-time spike trains.A decoder then selects the image class on the basis of the spike trains emitted by the output neurons.

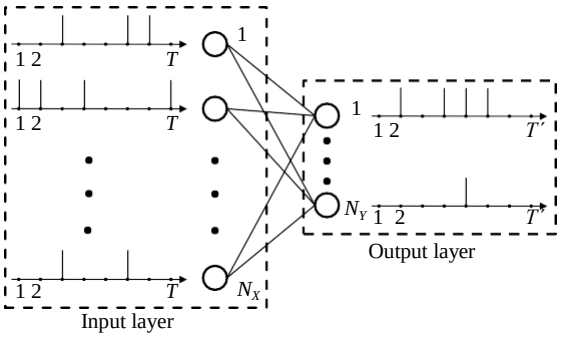


Fig. 1. Two-layer SNN for supervised learning.

Rate encoding. With the conventional rate encoding method, each entry of the input signal, e.g., each pixel for images, 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 entry. We use gray scale images with pixel intensities between 0 and 255 that yield a spike probability between 0 and 1⁄2.

GLM neuron model. The relationship between the input spike trains from the N_X presynaptic neurons and the output spike train of any postsynaptic neuron I follows a GLM

$$u_{i,t} = \sum_{j=1}^{N_X} \boldsymbol{\alpha}_{j,i}^T \mathbf{x}_{j,t-\tau_y}^{t-1} + \boldsymbol{\beta}_i^T \mathbf{y}_{i,t-\tau_y'}^{t-1} + \gamma_i,$$

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.

αj;i = A wj,i; βi = Bvi; A = [a_1, : : :, $a_{K\alpha}$] and B = [b_1, : : : , $b_{K\beta}$]

Kα, Kβ denote the respective number of basis functions.

a_k,b_k denote the raised cosine basis vectors.

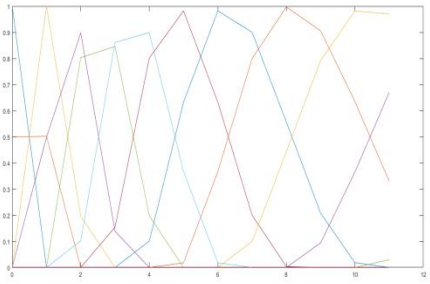


Fig. 2. Raised Cosine basis vectors

3. Training with First to Spike decoding

In this section, we introduce the proposed learning approach based on GLM neurons and 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 $P = g(u_i(t))$, 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 $p(t)$, summed over all time samples per simulation, T.

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

Some algebra gives the gradient with respect to the learnable parameters w and γ. (Not considering the feedback terms)

$$\nabla_{\mathbf{w}_{j,i}} L(\boldsymbol{\theta}) = \begin{cases} -\sum_{t=1}^{T} \rho_{i,t} h_t g(u_{i,t}) \mathbf{A}^T \mathbf{x}_{j,t-\tau_y}^{t-1} & i \neq c \\ -\sum_{t=1}^{T} \rho_{c,t} \left(h_t g(u_{c,t}) - q_t\right) \mathbf{A}^T \mathbf{x}_{j,t-\tau_y}^{t-1} & i = c \end{cases}$$

for the weights and

$$\nabla_{\gamma_i} L(\boldsymbol{\theta}) = \begin{cases} -\sum_{t=1}^{T} \rho_{i,t} h_t g(u_{i,t}) & i \neq c \\ -\sum_{t=1}^{T} \rho_{c,t} \left(h_t g(u_{c,t}) - q_t\right) & i = c \end{cases},$$

for the biases, where we have defined

$$\rho_{i,t} = \frac{g'(u_{i,t})}{g(u_{i,t})\,\bar{g}(u_{i,t})},$$

and

$$h_t = \sum_{t'=t}^{T} q_{t'} = 1 - \sum_{t'=1}^{t-1} q_{t'},$$

with

$$q_t = \frac{p_t(\boldsymbol{\theta})}{\sum_{t'=1}^{T} p_{t'}(\boldsymbol{\theta})}.$$

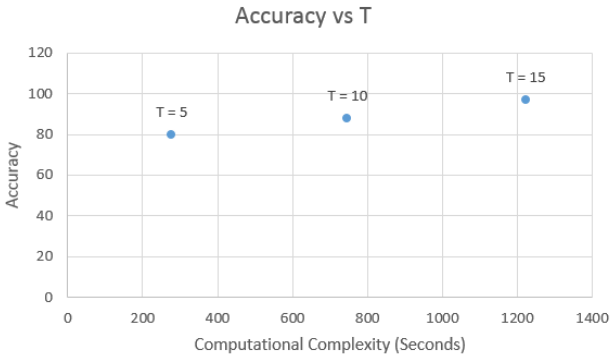
We now perform gradient descent to minimize the cost function.

4. Results

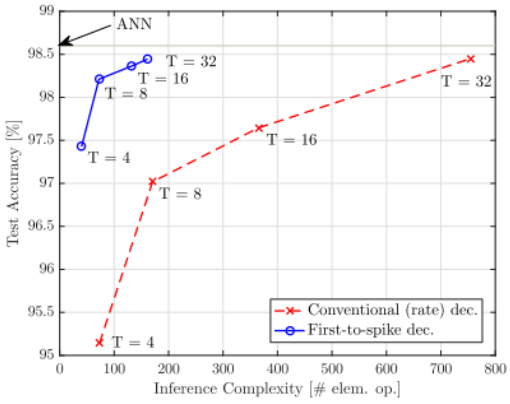
We have used a simplified dataset with two target classes, with four input neurons. Analogous to the MNIST encoding in the paper, the two classes correspond to input neuron spiking probabilities of (0.5,0.5,0.05,0.05) and (0.05,0.05,0.5,0.5).

The training is done on a dataset of 200 samples, while the accuracy is tested on a different dataset of 100 samples.

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.



Future work

In the next few weeks, we plan on implementing the same on MNIST dataset. Since using tanh activation function instead of the sigmoid gives superior results for ANNs, we can attempt to train the tanh-activated GLM model.

We are also looking at other possible probabilistic models.

References

Bagheri, A., Simeone, O., & Rajendran, B. (2017). Training Probabilistic Spiking Neural Networks with First-to-spike Decoding. Retrieved from http://arxiv.org/abs/1710.10704