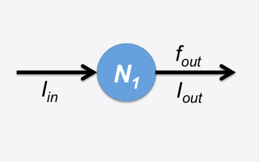
Arithmetic Computing via Rate Coding in Neural Circuits with Spike-triggered Adaptive Synapses

Sushrut Thorat, Bipin Rajendran
Indian Institute of Technology - Bombay

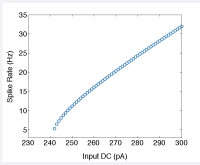
Arithmetic operations were previously implemented using neuronal pre-synaptic transfer functions. We implement them by controlling the neural network connectivity -> **Network topology**, and **Synaptic weight adaptation**. Our circuits encode and process information in the spike rates that lie between 40–140 Hz. The synapses in our circuit obey simple, local and spike-time dependent adaptation rules.

Linearisation: Spike Info (s_{out}) = Spike Rate (f_{out}) - 14.55

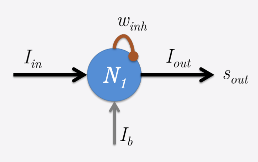
AEIF RS Neuron



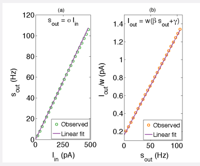
Spike rate response to input DC



Linearised Neuron

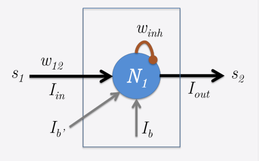


Spike rate response to input DC



Linear Operations:

SNN for scaling and offsetting



Spike rate response calculation

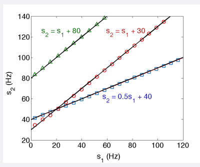
$$\langle I_{in} \rangle = w_{12}(\beta s_1 + \gamma)$$

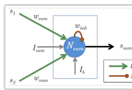
$$s_2 \approx \alpha(\langle I_{in} \rangle + I_b')$$

$$s_2 \approx w_{12}\alpha\beta s_1 + \alpha(I_b' + \gamma w_{12})$$

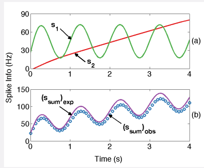
The relationship is approximate as I_{in} is not a DC current.

Spike rate response



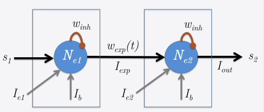


Addition of spike infos



Non-Linear Operations: $s_1^a s_2^b = \exp(a \log s_1 + b \log s_2)$

SNN for Exponentiation

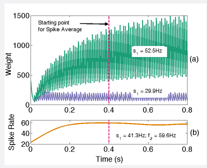


Exponentiation scheme

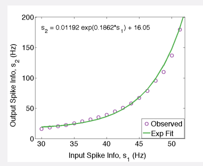
- Expected Super-linear response.
- w_{exp} should increase with increase in s_1.
- Weight Adaptation Rule:

$$w_{exp}(t+\Delta t) - w_{exp}(t) = (c' - a' w_{exp}(t))\,\Delta t + b' w_{exp}(t) \sum_{t^s} \delta(t - t^s)$$

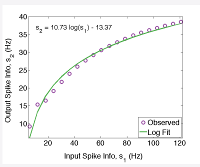
Weight adaptation

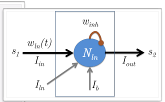


Exponential spike rate response



Logarithmic Spike rate response





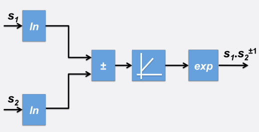
Weight Adaptation Rule:

$$w_{in}(t+\Delta t) - w_{in}(t) = (c - a w_{in}(t))\,\Delta t - b w_{in}(t) \sum_{t^s} \delta(t - t^s)$$

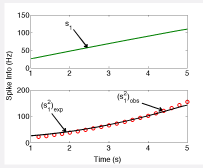
Multiplication Scheme

- Add logs of two spike rates.
- Match the range of log addition, and domain of exp, and implement exp.
- 0.5 X 0.1862 X 10.73 = 0.99

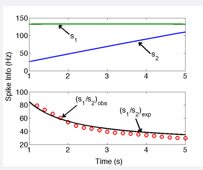
Multiplication SNN



Square of Spike Rate



Reciprocal of Spike Rate

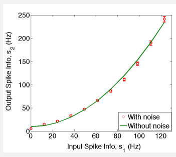


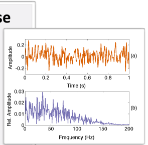
Other Operations

- $s_1^a s_2^b = \exp(a \log s_1 + b \log s_2)$
- We can theoretically implement any power law. The SNNs for Exp and Log have to be tuned appropriately.
- Using the linear operations, we can implement polynomial operations on spike trains.

Noise Resilience:

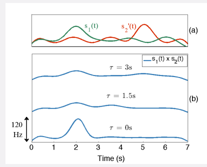
Multiplication response to Noise with CV 0.1

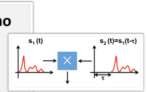




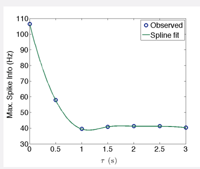
Application to Signal Estimation (Echolocation):

Multiplication of Signal and Echo





Max. Spike response curve



Conclusion:

- The building blocks we have designed in this paper can perform the fundamental operations – addition, subtraction, multiplication and division, as well as other non-linear transformations such as exponentiation and logarithm for time dependent signals in real-time, in presence of noise.
- Though our circuits use the AEIF neuron model, the outlined design methodology can be readily used to design SNN circuits that use other spiking neuron models.
- We have illustrated the power of these circuits to perform complex computations based on the frequency of spike trains in real-time, and thus they can be used in a wide variety of hardware and software implementations for navigation, control and computing.