A recurrent spiking neural network for prediction of non-linear dynamics

Hirak Jyoti Kashyap
kashyaph@uci.edu

June 07, 2017

1 The network dynamics

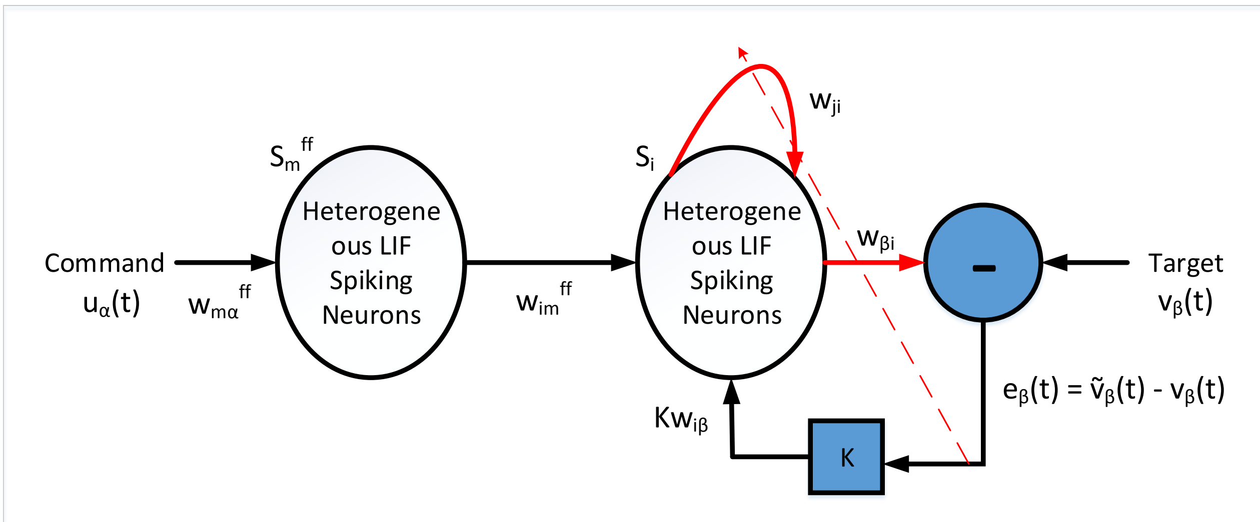

Figure 1: Architecture of the spiking neural network

Figure 1 depicts the architecture of the proposed spiking neural network. The architecture is similar to some existing recurrent neural networks proposed for learning and predicting non-linear dynamics, such as Echo state networks [1, 2], Liquid state machines, FORCE learning [3], and FOLLOW learning [4]. However, the proposed network differs from the existing literature in terms of biologically plausible LIF spiking activations and delta like local plasticity rule for learning.

Here, we briefly describe the dynamics of the network. The command u_α is a input signal(piece-wise linear/non-linear) that unambiguously signals an expected nonlinear target dynamic. This command is weighted by fixed normally distributed (mean = 1) set of weights (w_{ma}^{ff}) and delivered to each LIF neuron in Layer 1. The expression for incoming input to neuron m in Layer 1 at time t can be written as:

$$J_m^{ff} = \sum_\alpha (u_\alpha(t)) w_{ma}^{ff} \tag{1}$$

The spikes (S_m^{ff}) generated by the neurons in Layer 1 are propagated to Layer 2 using all-to-all feed-forward synapses (w_{im}^{ff}). These synapses are also weighted using a fixed normal distribution with mean 0.1. In addition to this feed-forward input, neuron i in Layer 2 also receives recurrent projections (w_{ij}) from all other neurons (j) in the same layer. However, these recurrent connections are plastic (shown in red) and updated using prediction error in each time step using a delta like learning rule. Finally, the neurons in Layer 2 also receives an error feedback at each time step through a gain modulated and fixed weighted connections (normally distributed with mean 0.2). The total input current to neuron i in layer 2 is given by the following equation.

$$J_i = \sum_m w_{im}^{ff}(S_m^{ff} * \kappa)(t) + \sum_j w_{ij}(S_j * \kappa)(t) + \sum_\beta K w_{i\beta}(e_\beta * \kappa)(t) \tag{2}$$

Where, κ is the non-linear exponential decay kernel (time constant 20 ms) and * is the convolution operator. K is a fixed negative gain on the error feedback, which has been set to -10 in our simulations.

The spikes of Layer 2 are accumulated using plastic readout connections with weights ($w_{\beta i}$) to generate the network prediction ($\tilde{v}_\beta$) at each time step.

$$\tilde{v}_\beta(t) = \sum_i w_{\beta i}(S_i * \kappa)(t) \tag{3}$$

From, this predicted output and the actual target signal ($v_\beta(t)$) at time t, the error is calculated as follows:

$$e_\beta(t) = \tilde{v}_\beta(t) - v_\beta(t) \tag{4}$$

1.1 Learning rule

We use a delta like local learning rules to update the recurrent weights and the output weights (shown in red). The updates are calculated as shown below.

Weight update for recurrent synapses:

$$w_{ij} = w_{ij} - \eta(\sum_\beta w_{i\beta} e_\beta)(S_j * \kappa)(t) \tag{5}$$

Weight update for readout synapses:

$$w_{\beta i} = w_{\beta i} - \eta(\sum_\beta w_{i\beta} e_\beta)(S_i * \kappa)(t) \tag{6}$$

References

[1] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *science*, 304(5667):78–80, 2004.

[2] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural computation*, 14(11):2531–2560, 2002.

[3] David Sussillo and Larry F Abbott. Generating coherent patterns of activity from chaotic neural networks. *Neuron*, 63(4):544–557, 2009.

[4] Aditya Gilra and Wulfram Gerstner. Predicting non-linear dynamics: a stable local learning scheme for recurrent spiking neural networks. *arXiv preprint arXiv:1702.06463*, 2017.