The Locust LGMD Neuron as a Deep Recurrent Neural Network

Abstract

Notes on correspondence between Gabbiani Lab's LGMD Neuron model and a deep RNN model.

1 A Model of the Locust LGMD Neuron

The locust LGMD neuron plays a central role in looming detection and the generation of escape responses, a behavior critical for a locust's survival. Here we briefly review the latest biophysical model of the locust LGMD neuron [Richard Dewell, Gabbiani Lab]. The model divides the dendrites and LGMD neuron surface into a set of compartments $i \in \mathcal{I}$ and describes the dynamics of the membrane potential within each compartments by a system of ODEs derived from Kirchhoff's current conservation law:

$$I_{\text{capacitive}} \equiv C\dot{V}_i = I_{\text{leak}} + I_{\text{axial}} + I_{\text{syn}} + I_{\text{chan}} \tag{1}$$

$$\equiv g_L(V - E_L) + \sum_{j \in \mathcal{N}_i} g_{ij}(V_i - V_j) + \bar{g}_{\text{syn}}(t)(V_i - E_{\text{syn}}) + \sum_{c \in \mathcal{C}_i} g_c(V_i, \bar{V}_i)s_c(V_i - E_c), \tag{2}$$

where $V_i(t)$ is the voltage for compartment i, C is the membrane capacitance, the g's are conductances, the E's are reversal potentials, and $\mathcal{C}_i$ is the set of all channels present in compartment i.

Let's consider the role of each current in turn. This will help us later when we attempt to establish a correspondence with terms in a recurrent neural network (RNN).

Leak and Axial Currents The leak current, I_{leak}, alone would drive each compartment $V_i(t) \to E_L$ over the timescale $\tau_L \equiv g_L/C$. The axial current I_{axial} will work to equalize the voltages in neighboring compartments, flattening out any non-uniformities in the potentials. In combination with I_{leak}, this would drive the system to its resting potential $V_i(t) \to E_L$ even faster. A corollary is that if the input is constant in time $x_t = \text{const}$, the conductance model will have a transient response which quickly converges to the resting potential. Note that both currents have conductances that are *independent* of the input.

Synaptic Current In contrast, the synaptic current I_{syn}'s conductance is **input-dependent** and therefore dynamic:

$$\bar{g}_{\text{syn}}(t) \equiv \kappa(t) \star S_{\text{syn}}(t) \tag{3}$$

$$\approx EMA_{\tau_s}[S_{\text{syn}}(t)] - EMA_{\tau_l}[S_{\text{syn}}(t)] \tag{4}$$

$$S_{\text{syn}}(t) \equiv \sum_{e \in \mathcal{E}_i} \delta(t - t_e^*) \tag{5}$$

where $e \in \mathcal{E}_i$ indexes all synaptic (firing) events in compartment i, $S(t)$ is the stream of presynaptic firing events and t_e^* is the time of event e. $\kappa(t) \equiv t\exp(-t/\tau_{\text{syn}})$ is a bump kernel that is convolved with the input presynaptic event stream $S(t)$, and whose action can be approximated by the difference of two EMAs with short/long time constants. The EMA operator implements an exponential moving average (EMA) in time with timescale $\tau_{\text{evt}} \approx 20$ ms, but can vary in range from $\tau_s \approx 1/10$ ms to $\tau_l \approx 2,000$ ms. Therefore, the synaptic conductance serves to smoothen the input synaptic event stream, effectively reducing noise and allowing multiple asynchronous events to combine together to exceed threshold (otherwise synaptic events would have to be highly synchronized in order to have any effect downstream).

Channel Current I_{chan} is the only term with that feature the application of a nonlinearity. The channel conductance $g_c \equiv g_c(V(t), m_c(t), n_c(t))$ depends on the voltage and also on the channel receptor binding state, which is composed of the activation state $m_c(t) \in \mathbb{R}_+$ and the inactivation state $n_c(t) \in \mathbb{R}_+$. Both of these measure the density of channels (number per unit surface area) that are activated (opened) during a synaptic firing event. Mathematically, both are modeled as continuous-time Markov chains:

$$\dot{m}_{ic}(t) = -\frac{1}{\tau_m}\left(m_{ic}(t) - m_{ic,\infty}(V_i(t))\right) \tag{6}$$

$$\dot{n}_{ic}(t) = -\frac{1}{\tau_n}\left(n_{ic}(t) - n_{ic,\infty}(V_i(t))\right) \tag{7}$$

$$m_\infty(V) \equiv \sigma(W_m V + b_m) \tag{8}$$

$$n_\infty(V) \equiv \sigma(-W_n V - b_n) \tag{9}$$

Then the total channel conductance is

$$g_{ic}(t) = g_{ic}^{\max} \cdot m_{ic}(t)^{p_c} \cdot n_{ic}(t)^{q_c}, \tag{10}$$

where $g_{ic}^{\max}$ is the maximal channel conductance and

2 Comparison with Vanilla RNNs

Given these properties of the currents, we now attempt to establish a correspondence with RNNs.

Leak and Diffusion Currents as Hidden State-Dependent Terms The leak and diffusion currents $I_{\text{leak}} + I_{\text{axial}}$ must correspond to the linear hidden state update $h_{t+1} - h_t = Wh_t$ in an RNN, where the weight matrix W has diagonal elements $W_{ii} \equiv -1/\tau_L = -g_L/C$ and off-diagonal elements $W_{ij} = -[i \sim j]g_{ij}/C$, where the relation $i \sim j$ holds if compartments i, j are neighbors.

Synaptic Conductance as Multiplicative Input-Dependent Gating of Hidden State The synaptic current I_{syn} is the only *input-dependent* current and thus roughly corresponds to the input-dependent term in the RNN $\Delta_t h_t = Ux_t$. However, note that it differs in three important ways:

- Unlike in an RNN where it takes in the raw input, the conductance model actually smoothens the input with an EMA.
- Unlike in the RNN, where there is a global clock (time $t \in \mathbb{N}$ is discrete), in the conductance model there is no such clock (time $t \in \mathbb{R}$ is continuous).
- Unlike an RNN where the input multiplies a weight matrix Ux_t, in the conductance model the (smoothened) input *multiplies* the hidden state. In the RNN this would be equivalent to a **multiplicative** term $h_t \odot \bar{x}_t$ term.

Channel Conductance as Nonlinearity The channel currents I_{chan} are the only nonlinear terms in the conductance model. Hence one might anticipate they will correspond to the nonlinear update in an RNN.

However, again we see some fundamental differences with vanilla RNNs:

- The channel conductance is not dependent on the input(!). Furthermore, it is a nonlinear (sigmoidal or bump-shaped) function of the hidden state *per channel*.
- The nonlinearity is not composed over time. Instead it adds linearly to the hidden state $V(t)$. This prevents chaotic behavior i.e. exploding or vanishing activations.

- The hidden state at any time is a compromise between several attractors $V_i \in \{E_L, E_{\mathrm{syn}}, E_{\mathrm{Na}}, E_{\mathrm{K}}, E_{\mathrm{Cl}}, E_{\mathrm{Ca}}\}$. This keeps the hidden state contained within a bounded region, preventing exploding/vanishing while also maintaining a useful dynamic range.

We will preserve these features and define a new kind of RNN in the next section.

3 Charged RNNs: A New Kind of RNN based on Conductance Models of Real Neurons

Combining the observations above, we define the **Charged RNN (cRNN)** as follows:

$$\Delta_t h_t = -W(G)(h_t - E_L) + (h_t - E_S) \odot \bar{x}_t + \sigma_+(h_t) \odot \sigma_-(h_t) \odot (h_t - E_c)$$

We can also define a simpler variant that relaxes some of the structure from the conductance-based model:

$$\Delta_t h_t = -W h_t + h_t \odot \bar{x}_t + \sigma_+(h_t) \odot \sigma_-(h_t) \odot h_t + b,$$

where we absorb all of the terms dependent on the reversal potentials into a bias term.

3.1 Comparison with Vanilla and LSTM RNNs

Charged RNNs possess several important properties that differentiate them from traditional vanilla and LSTM RNNs.

Event-Driven When the input is constant in time, the cRNN converges to its resting state. Also, we can imagine a continuous time event-driven implementation that takes asynchronous event streams as input rather than globally clocked ones.

Stability due to Learned Parameter Constraints The learned parameters in a cRNN are the channel densities, which are highly constrained: they are intrinsically nonnegative and bounded by the membrane surface area of the compartment.

Dynamic Range Utilization due to Fixed Parameter Constraints All other (unlearned) parameters include the membrane capacitance, non-channel (leak and synaptic) conductances, the reversal potentials, any timescale constants. As these are fixed, they cannot contribute to any learning. However, they do have the benefit of keeping the dynamic range of the neuron fixed and useful. In other words, the hidden state (current) is bounded within a range defined by the reversal potentials, and is at any given time a compromise between all of the reversal potentials.

Stability due to Multiplicative Gating of Input Due to the input-dependent gating term, the derivatives computed in the backpropagation algorithm can be controlled dynamically by the input. This allows any exploding or vanishing eigenvalues in the weight matrices to be compensated for dynamically.

Stability due to Additive Non-Compositional Nonlinearity Traditional RNNs employ a composition of nonlinearities, typically sigmoidal or hyperbolic tangent. When composed repeatedly, these nonlinearities can lead to chaotic or unstable behavior i.e. the exploding/vanishing activations problem. But cRNNs are far less prone to this problem due to their additive nonlinearity, which only *adds* to the prior hidden state. Thus the nonlinearity is far more controlled

We conjecture that the many forms of stabilization shown above combine together to yield far more stable behavior during inference and learning. In the next section we test these hypotheses experimentally.

4 Experimental Results

References