Notes on Boerlin, Machens and Deneve

Driven Network

Suppose that we want to construct a spiking network to accurately encode a function $f(t)$. The response of neuron i, where $i = 1, 2, \ldots, N$, is denoted by

$$\delta_i(t) = \sum_\mu \delta(t - t_i^\mu), \tag{1}$$

where $\delta()$ is a Dirac δ function, and t_i^μ are the spike times for neuron i. We also define an integrated "synaptic current" produced by this spike train, s_i, defined by

$$\tau \dot{s}_i = -s_i + \tau \delta_i, \tag{2}$$

where the dot denotes a time derivative. Note that

$$s_i \rightarrow s_i + 1 \tag{3}$$

when neuron i spikes and that s_i decays exponentially otherwise. The signal is decoded as

$$z = \sum_{i=1}^N w_i s_i, \tag{4}$$

and the goal is to make $z(t)$ as close to $f(t)$ as possible. From equation 2,

$$\tau \dot{z}_i = -z_i + \tau \sum_{i=1}^N w_i \delta_i. \tag{5}$$

The idea of BMD is that neuron i should fire an action potential at time t if its firing would reduce the error $(f(t) - z(t))^2$. From equations 3 and 5, $z \rightarrow z + w_i$ when neuron i spikes. Thus, the condition for firing is

$$(f - z - w_i)^2 < (f - z)^2 \quad \text{or} \quad w_i(f - z) > \frac{w_i^2}{2}. \tag{6}$$

This leads BMD to conclude that

$$v_i = w_i(f - z) \tag{7}$$

can be identified as a measure of the membrane potential of a neuron and

$$T_i = \frac{w_i^2}{2} \tag{8}$$

can be identified as a measure of its firing threshold. Taking the time derivative of equation 7 and using equation 5, we find

$$\begin{aligned}
\tau\dot{v}_i &= w_i(\tau\dot{f} - \tau\dot{z}) = w_i\left(\tau\dot{f} + z - \tau\sum_{j=1}^{N} w_j\delta_j\right) \\
&= -w_i(f - z) + w_i(\tau\dot{f} + f) - \tau w_i\sum_{j=1}^{N} w_j\delta_j \\
&= -v_i + w_i(\tau\dot{f} + f) - \tau w_i\sum_{j=1}^{N} w_j\delta_j\,. \qquad (9)
\end{aligned}$$

This and the condition for firing

$$v_i > T_i = \frac{w_i^2}{2} \qquad (10)$$

describe a network of integrate-and-fire neurons with fast (instantaneous) synaptic connections described by $J_{ij}^{\text{fast}} = -\tau w_i w_j$ and an input current to neuron i given by $w_i(\tau\dot{f} + f)$. Note that, according to equation 9, when neuron j fires

$$v_i \rightarrow v_i - w_i w_j\,. \qquad (11)$$

In particular, when $i = j$, this produces the reset

$$v_i \rightarrow v_i - w_i^2\,. \qquad (12)$$

This means that neuron i fires when $v_i = T_i$ and is then reset to $v_i = -T_i$.

There is one modification that needs to be made to deal with the case when w_i is very small. In this case, neuron i has little effect on z, but the above equations would give it a very small threshold, producing firing at a very high rate. To prevent this, BMD introduce a parameter μ and change the threshold to

$$v_i > T_i = \frac{1}{2}(w_i^2 + \mu)\,. \qquad (13)$$

and the network equation is changed to

$$\tau\dot{v}_i = -v_i + w_i(\tau\dot{f} + f) - \tau w_i\sum_{j=1}^{N} w_j\delta_j - \tau\mu\delta_i\,. \qquad (14)$$

This assures that the neuron is again reset to $v_i = -T_i$, with T_i given by equation 17. It also implies that $J_{ij}^{\text{fast}} = -\tau(w_i w_j + \mu\delta_{ij})$.

These results can be extended to the case of multiple encoded functions $f^a(t)$ for $a = 1, 2, \ldots, P$. These signals are encoded as

$$z^a = \sum_{i=1}^{N} w_i^a s_i\,. \qquad (15)$$

In this case, the equations of the network are

$$\tau\dot{v}_i = -v_i + \sum_{a=1}^{P} w_i^a(\tau\dot{f}^a + f^a) - \tau\sum_{a=1}^{P} w_i^a \sum_{j=1}^{N} w_j^a\delta_j - \tau\mu\delta_i\,. \tag{16}$$

with spiking for

$$v_i > T_i = \frac{1}{2}\left(\sum_{a=1}^{P}(w_i^a)^2 + \mu\right). \tag{17}$$

This means that, now, $J_{ij}^{\text{fast}} = -\tau(\sum_a w_i^a w_j^a + \mu\delta_{ij})$

Autonomous Network

The network in the previous section is driven by the input $\tau\dot{f} + f$ so that it can encode the signal f. We now want construct an autonomous system that encodes f without requiring this input. To do this, we supppose that the functions $f^a(t)$ are generated by a dynamic system of the form

$$\tau\dot{f}^a = -f^a + \sum_{b=1}^{P} B^{ab}F(f^b) + I^a\,. \tag{18}$$

Two typical cases are $F(f) = f$ and $F(f) = \tanh(f)$. This equation implies that we can replace equation 16 by

$$\tau\dot{v}_i = -v_i + \sum_{a=1}^{P} w_i^a \sum_{b=1}^{P} B^{ab}F(f^b) - \tau\sum_{a=1}^{P} w_i^a \sum_{j=1}^{N} w_j^a\delta_j - \tau\mu\delta_i + \sum_{a=1}^{P} w_i^a I^a\,. \tag{19}$$

These equations still imply an external input that specifies f. We now consider two cases.

<u>Case 1 - Linear F:</u> If $F(f) = f$, we can use the approximation $f \approx z$ to write

$$\tau\dot{v}_i = -v_i + \sum_{a=1}^{P} w_i^a \sum_{b=1}^{P} B^{ab} \sum_{j=1}^{N} w_j^a s_j - \tau\sum_{a=1}^{P} w_i^a \sum_{j=1}^{N} w_j^a\delta_j - \tau\mu\delta_i + \sum_{a=1}^{P} w_i^a I^a\,. \tag{20}$$

Now, along with the J_{ij}^{fast} written above, we have $J_{ij}^{\text{slow}} = \sum_{a,b} w_i^a B^{ab} w_j^b$.

<u>Case 2 - Nonlinear F:</u> In this case, we can use a learning procedure, such as FORCE learning, to find a set of coefficients i_i^a such that

$$\sum_{i=1}^{N} u_i^a s_i \approx \sum_{b=1}^{N} B^{ab}F(f^b)\,. \tag{21}$$

We then write

$$\tau\dot{v}_i = -v_i + \sum_{a=1}^{P} w_i^a \sum_{j=1}^{N} u_j^a s_j - \tau\sum_{a=1}^{P} w_i^a \sum_{j=1}^{N} w_j^a\delta_j - \tau\mu\delta_i + \sum_{a=1}^{P} w_i^a I^a\,, \tag{22}$$

which means that $J_{ij}^{\text{slow}} = \sum_a w_i^a u_j^a$, with u_j^a learned.