#DOC421 - Study Notes for Computational Neurodynamics

Paul Gribelyuk (pg1312, a5)

April 30, 2013

1 Numerical Simulation

Assume that:

$$\frac{dy}{dt} = f(y)$$

- Euler Method

$$y(t+\delta t) \approx y(t) + \delta t f(y(t))$$

- Runge-Kutta Method:

$$y(t+\delta t) \approx y(t) + \frac{1}{6}\delta t(k_1 + 2k_2 + 2k_3 + k_4)$$

  where:

$$\begin{aligned} k_1 &= f(y(t)) \\ k_2 &= f(y(t) + \frac{1}{2}\delta t k_1) \\ k_3 &= f(y(t) + \frac{1}{2}\delta t k_2) \\ k_4 &= f(y(t) + \delta t k_3) \end{aligned}$$

2 Neuron Firing Models

- Hodgkin-Huxley

$$\boxed{C\frac{dv}{dt} = -\sum_k I_k + I_{ext}(t)}$$

where:

$$
\begin{aligned}
\sum_k I_k &= g_{Na} m^3(v) h(v)(v - E_{Na}) + g_K n^4(v)(v - E_k) + g_L(v - E_L) \\
\frac{dm}{dt} &= \alpha_m(v)(1 - m(v)) - \beta_m(v) m(v) \\
\frac{dn}{dt} &= \alpha_n(v)(1 - n(v)) - \beta_n(v) n(v) \\
\frac{dh}{dt} &= \alpha_h(v)(1 - h(v)) - \beta_h(v) h(v) \\
\alpha_m(v) &= \frac{2.5 - 0.1v}{e^{2.5-0.1v} - 1} \qquad \beta_m(v) = 4e^{-\frac{v}{18}} \\
\alpha_n(v) &= \frac{0.1 - 0.01v}{e^{1-0.1v} - 1} \qquad \beta_n(v) = 0.125e^{-\frac{v}{80}} \\
\alpha_h(v) &= 0.07e^{-\frac{v}{20}} \qquad \beta_h(v) = \frac{1}{e^{3-0.1v} + 1} \\
g_{Na} &= 120 \qquad g_K = 36 \qquad g_L = 0.3 \\
E_{Na} &= 115 \qquad E_K = -12 \qquad E_L = 10.6
\end{aligned}
$$

- Leaky Integrate and Fire (LIF)

$$
\begin{aligned}
\tau \frac{dv}{dt} &= v_r - v + R \cdot I_{ext}(t) \\
\text{if } v &\geq \theta \text{ then } v \leftarrow v_r
\end{aligned}
$$

where:

$$\tau = 5 \qquad R = 1 \qquad v_r = -65mV \qquad \theta = -50mV$$

we insert a spike before the volate reset. To address a mandatory refactory period, we can disallow recording a spike until some time α after the time of the last spike t_{spike}:

$$\text{if } \{v \geq \theta \text{ and } t - t_{spike} > \alpha\} \text{ then } \{v \leftarrow v_r \text{ and } t_{spike} \leftarrow t\}$$

- Quadratic Integrate and Fire

$$\tau \frac{dv}{dt} = a(v_r - v)(v_c - v) + R \cdot I_{ext}(t)$$

- Izhikevich

$$
\begin{aligned}
\frac{dv}{dt} &= 0.04v^2 + 5v + 140 - u + I_{ext}(t) \\
\frac{du}{dt} &= a(bv - u)
\end{aligned}
$$

$$\text{if } v \geq 30 \text{ then } \{v \leftarrow v_r \text{ and } u \leftarrow u + d\}$$

For excitatory neurons: $a = 0.02 \quad b = 0.2 \quad c = -65 \quad d = 8$
For inhibitory neurons: $a = 0.02 \quad b = 0.25 \quad c = -65 \quad d = 2$
For bursting neurons: $a = 0.02 \quad b = 0.25 \quad c = -55 \quad d = 0$

3 Small-World Networks

A *network* is a graph $G = \langle V, E \rangle$ of nodes in V and edges in $E \subseteq V \times V$. We have a connectivty matrix A where $A(i,j) = \delta_{(j,i)\in E}$. For directed networks $A(i,j) = A(j,i)$ also holds. There are no self-connection, so $A(i,i) = 0$. The *degree* of a node i, k_i, is the number of edges it is part of. The average degree of an undirected graph with n nodes and m edges is:

$$k = \frac{2m}{n}$$

Random networks have a fixed probabilty p of any two nodes being connected. The small-world index of a graph G is defined:

$$\sigma_G = \frac{\gamma_G/\gamma_{rand}}{\lambda_G/\lambda_{rand}} = \frac{\gamma_G}{\lambda_G} \Big/ \frac{\gamma_{rand}}{\lambda_{rand}}$$

where λ is the average *path length*, and γ is the *clustering coefficient* To create a small-world network, can use the Watts-Strogatz method:

1 Create ring lattice of degree k

2 With probability, p, re-wire an un-rewired edge to any other node in the network.

Global efficiency of a network G can be measured:

$$Eff_{glob}(G) = \frac{1}{n(n-1)} \sum_{i \neq j} Eff(i,j)$$

where $Eff(i,j) = \frac{1}{\lambda(i,j)}$. *Local efficiency* is the average of all global efficiencies of subnetworks, G_i corresponding to the direct neighbors of nodes. This can be measured:

$$Eff_{loc}(G) = \frac{1}{n} \sum_{i \in G} Eff_{glob}(G_i)$$

4 Modular Networks

We can generate modular networks with n nodes and m edges by creating C communities where each community has n/C nodes and m/C random edges between nodes in the community. We the randomly rewire intra-community edges to be intercommunity edges with probability p. We can take a graph G and a partitioning P of that graph into communities and measure its *modularity*:

$$Q(P) = \sum_c \sum_{i,j \in c} \frac{A(i,j)}{2m} - \frac{k_i}{2m} \frac{k_j}{2m} = \frac{1}{2m} \sum_{i,j} \left(A(i,j) - \frac{k_i k_j}{2m} \right) \delta_{c_i c_j}$$

where c_i corresponds to the community of node i, and δ_{xy} is the delta function. We can spacially embed networks by allowing the probability of a connection between two nodes to vary with their mutual distance:

$$P(A(i,j) = 1) = e^{-hd(i,j)}$$

where h is a pre-defined constant. As h increases, σ and Q increase. *Hub nodes* are nodes which represent the majority of inter-module connections. The *participation index* of a node can be defined:

$$P_i = 1 - \sum_c \left(\frac{k_i^c}{k_i} \right)^2$$

for i to be a connector node, $k_i > k$ and $P_i > 0.3$.

5 Dynamical Complexity

For an input set S of N time-series (mean-firing rates of neurons), we can define quantities:

- Entropy
$$H(S) = \frac{1}{2} \ln \left((2\pi e)^N \det(COV(S)) \right)$$
- Mutual Information
$$MI(X, S - \{X\}) = H(X) + H(S - \{X\}) - H(S)$$
- Integration
$$I(S) = \sum_{i=1}^{N} H(X_i) - H(S)$$
- Complexity
$$C(S) = \sum_{i=1}^{N} MI(X_i, S - \{X\}) - I(S)$$
High segregation means low mutual information. Overly high integration means MI terms are high but so is the $I(S)$ term. A balance between segregation and integration means MI term can be high while the $I(S)$ term remains low.
- Granger Causality: For every time-series in S, model:
$$X_i(t) = \sum_j \sum_{n=1} NAjX_n(t-j) + B_j X_n(t-j) + C_j X_n(t-j) + \epsilon_N(i,t)$$
and without some term X_{n_0}:
$$X_i(t) = \sum_j \sum_{n \neq n_0} AjX_n(t-j) + B_j X_n(t-j) + C_j X_n(t-j) + \epsilon_{n_0}(i,t)$$
if variance of $\epsilon_N(i,t) \ll \epsilon_{n_0}(i,t)$, then X_{n_0} *Granger-causes* X_i. Causal density can be defined:
$$\alpha / n(n-1)$$
where α is number of pairs (i, j) where X_i Granger-causes X_j.
- Coalition Entropy
$$H_C = -\frac{1}{\log_2 |L|} \sum_{s \in L} p(s) \log_2(p(s))$$

6 Synchrony

There are 4 frequency bands of neuronal firing: Theta (4-8Hz), Alpha (8-15Hz), Beta (15-30Hz), and Gamma (30-80Hz).

- Extracting synchrony from mean firing rates We can extract phase information from a time series $X(t)$:

  $$\xi(t) = X(t) + iX_H(t) = A(t)e^{i\theta(t)}$$

  To calculate the Hilbert transform $X_H(t)$:

  $$X_h(t) = \frac{1}{\pi} P.V. \int_{\mathbb{R}} \frac{X(\tau)}{t-\tau} d\tau$$

  Then the instantanious phase is:

  $$\theta(t) = \arctan\left(\frac{X_H(t)}{X(t)}\right)$$

  The synchrony of a community of oscillators is:

  $$\phi_c(t) = \left| \left\langle e^{i\theta_k(t)} \right\rangle_{k \in c} \right|$$

  is the norm of average of the complex representation of the phase of each oscillator in the community.

- Kuramoto Oscillators

  $$\frac{d\theta_i}{dt} = \omega_i + \frac{1}{N+1} \sum_{j=1}^{N} K_{i,j} \sin(\theta_j - \theta_i - \alpha)$$

  where ω_i is natural frequency, $K_{i,j}$ is coupling strength, and α is phase lag. Chimera states arise when α is slightly less than $\frac{\pi}{2}$.

- Chimera and Metastability

  $$\begin{aligned}
  \sigma_{chi}(t) &= \frac{1}{M-1} \sum_{c \in C} (\phi_c(t) - \langle \phi(t) \rangle_C)^2 \\
  \sigma_{met}(c) &= \frac{1}{T-1} \sum_{t \le T} (\phi_c(t) - \langle \phi(t) \rangle_T)^2 \\
  \chi &= \langle \sigma_{chi} \rangle_T \\
  \lambda &= \langle \sigma_{met} \rangle_C
  \end{aligned}$$

7 Plasticity

Updating weights between neurons (STDP):

$$\Delta\omega = \begin{cases} A^+ e^{-\Delta t/\tau^+} & \text{if} \quad \Delta \ge 0 \\ -A^- e^{\Delta t/\tau^-} & \text{if} \quad \Delta < 0 \end{cases}$$