Course Summary

• Where have we been?
  • Course Highlights

• Where do we go from here?
  • Challenges and Open Problems

• Further Reading
What is the neural code?

What is the nature of the code? Representing the spiking output:
- single cells vs populations
- rates vs spike times vs intervals
What features of the stimulus does the neural system represent?

Encoding and decoding neural information

Encoding: building functional models of neurons/neural systems and predicting the spiking output given the stimulus

Decoding: what can we say about the stimulus given what we observe from the neuron or neural population?
Key concepts: Poisson & Gaussian

Spike trains are variable
Models are probabilistic
Deviations are close to independent

Highlights: Neural Encoding

stimulus $X(t)$ → spike-triggering stimulus features $f_1, f_2, f_3$ → multidimensional decision function $x_1, x_2, x_3$ → spiking output $\rho(t)$
Highlights: Finding the feature space of a neural system

Highlights: Finding an interesting tuning curve
Decoding: Signal detection theory

Decoding corresponds to comparing test to threshold.
\[ \alpha(z) = P[r \geq z | -] \] false alarm rate, “size”
\[ \beta(z) = P[r \geq z | +] \] hit rate, “power”

Highlights: Neurometric curves

A

Fraction correct vs. coherence (%)

B

Number of trials vs. coherence

Firing rate (Hz) vs. coherence
Decoding from a population
e.g. cosine tuning curves

\[ \overline{\mathbf{t}}_{\text{pop}} = \sum_{a=1}^{4} \left( \frac{r}{r_{\text{max}}} \right) \overline{c}_a \]

RMS error in estimate

More general approaches: MAP and ML

**MAP:**  \( s^* \) which maximizes \( p[s|r] \)

**ML:**  \( s^* \) which maximizes \( p[r|s] \)

Difference is the role of the prior: differ by factor \( p[s]/p[r] \)

For cercal data:
Highlights:
Information maximization as a design principle of the nervous system

The biophysical basis of neural computation
Excitability is due to the properties of ion channels

- Voltage dependent
- Transmitter dependent (synaptic)
- Ca dependent

Highlights: The neural equivalent circuit

Ohm's law: \( V = IR \) and Kirchhoff's law

- \( -C_m \frac{dV}{dt} = \sum g_i(V - E_i) + I_e \)
Simplified neural models

A sequence of neural models of increasing complexity that approach the behavior of real neurons

*Integrate and fire neuron:*  
  - subthreshold, like a passive membrane  
  - spiking is due to an imposed threshold at $V_T$

*Spike response model:*  
  - subthreshold, arbitrary kernel  
  - spiking is due to an imposed threshold at $V_T$  
  - postspike, incorporates afterhyperpolarization

*Simple model:*  
  - complete 2D dynamical system  
  - spiking threshold is intrinsic  
  - have to include a reset potential

---

Simplified models: integrate-and-fire

\[ \tau_m \frac{dV}{dt} = -(V - E_L) + I_e R_m \]

If $V > V_{\text{threshold}} \Rightarrow \text{Spike}$

Then reset: $V = V_{\text{reset}}$

Integrate-and-Fire Model
Simplified models: spike response model

Gerstner; Keat et al. 2001

Highlights: Dendritic computation

Filtering
Shunting
Delay lines
Information segregation
Synaptic scaling
Direction selectivity
Highlights: Compartmental models

Neuronal structure can be modeled using electrically coupled compartments

Connecting neurons: Synapses

Presynaptic spikes cause neurotransmitters to cross the cleft and bind to postsynaptic receptors, allowing ions to flow in and change postsynaptic potential
EPSPs and IPSPs

Size of PSP is a measure of synaptic strength. Can vary on the short term due to input history. Long term due to synaptic plasticity (LTP/LTD).

Networks
Modeling Networks of Neurons

\[ \tau \frac{dv}{dt} = -v + F(Wu + Mv) \]

Output  Decay  Input  Feedback

Highlights: Unsupervised Learning

- For linear neuron: \( v = w^Tu = u^Tw \)
- **Basic Hebb Rule**: \( \tau_w \frac{dw}{dt} = uv \)
- Average effect over many inputs: \( \tau_w \frac{dw}{dt} = \langle uv \rangle = Qw \)
- \( Q \) is the input correlation matrix: \( Q = \langle uu^T \rangle \)
Highlights: Generative Models

Droning lecture  Lack of sleep  Mathematical derivations

Highlights: Generative Models and the Connection to Statistics

Unsupervised learning = learning the hidden causes of input data

\( p[v; G] \) (prior)  Causes  \( p[v | u; G] \) (posterior)

Generative model

Use EM algorithm for learning the parameters \( G \)

\( p[u | v; G] \) (data likelihood)

\( G = (\mu_v, \sigma_v) \)

Causes of clustered data
Highlights: Supervised Learning: Neurons as Classifiers

Perceptron:

Inputs $u_j$ (-1 or +1)

Weighted Sum

Threshold

Output $v_i$ (-1 or +1)

Separating hyperplane:

Highlights: Supervised Learning: Regression

Backpropagation for Multilayered Networks

Finds $W$ and $w$ that minimize errors:

$$E(W_{ij}, w_{jk}) = \frac{1}{2} \sum_{m,l} (d_{ij}^m - v_i^m)^2$$

Desired output

Example: Truck backer upper
Highlights: Reinforcement Learning

• Learning to predict **delayed rewards** (TD learning):
  \[ w(\tau) \rightarrow w(\tau) + \varepsilon [r(t) + v(t + 1) - v(t)] u(t - \tau) \]

• Actor-Critic Learning:
  • Critic learns value of each state using TD learning
  • Actor learns best actions based on value of next state (using the TD error)

---

The Future: Challenges and Open Problems

• How do neurons encode information?
  • **Topics**: Synchrony, Spike-timing based learning, Dynamic synapses

• How does a neuron’s structure confer computational advantages?
  • **Topics**: Role of channel dynamics, dendrites, plasticity in channels and their density

• How do networks implement computational principles such as **efficient coding** and **Bayesian inference**?

• How do networks learn “optimal” representations of their environment and engage in **purposeful behavior**?
  • **Topics**: Unsupervised/reinforcement/imitation learning
Further Reading (for the summer and beyond)

- *The Biophysics of Computation*, C. Koch, Oxford University Press, 1999
- *Large-Scale Neuronal Theories of the Brain*, C. Koch and J. L. Davis, MIT Press, 1994
- *Bayesian Brain*, K. Doya et al., MIT Press, 2007

Next meeting: Project presentations!

- Project presentations will be on **Monday, June 10, 10:30am-12:20pm in the same classroom**
- Keep your presentation short: ~8 slides, 8 mins/group
- Slides:
  - Bring your slides on a USB stick to use the class laptop
  - OR
  - Bring your own laptop if you have videos etc.
- Projects reports (10-15 pages total) due by **midnight Tuesday, June 11** (by email to both Adrienne and Raj)
Have a great summer!

Au revoir!