

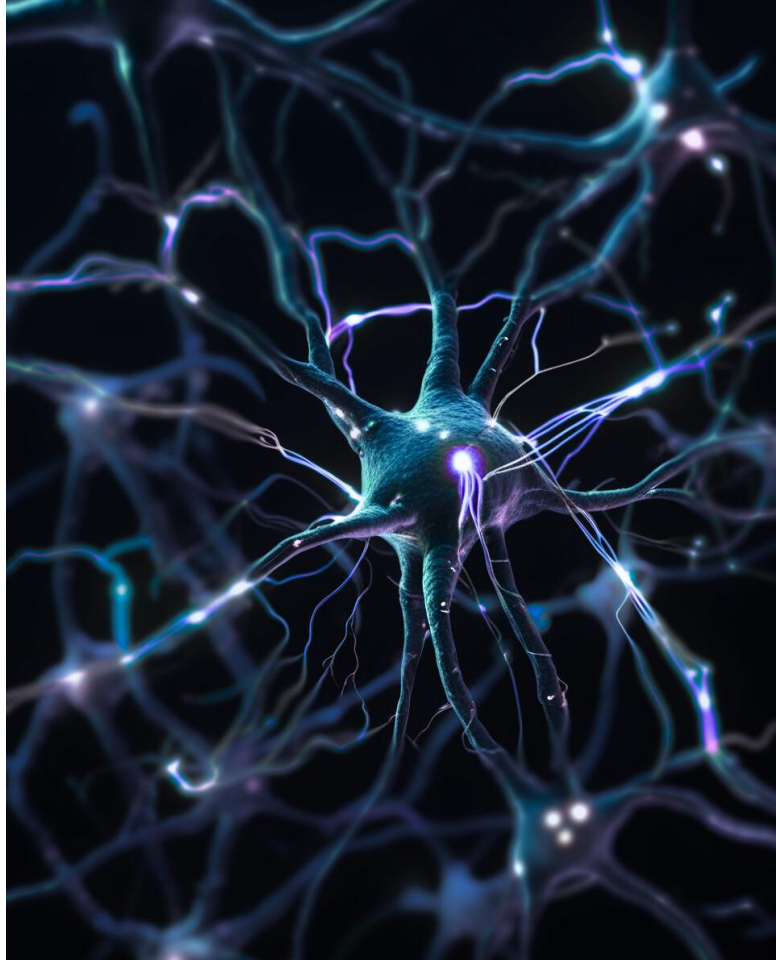


Computational Neuroscience

Session 0: Introductions

Instructor: Ashkan Damavandi

July 30, 2025



Session Outline

1 What is Computational Neuroscience?

2 Key Papers

2.1 Hodgkin & Huxley (1952): Quantitative description of membrane current

2.2 David Marr (1969): A theory of cerebellar cortex

2.3 John Hopfield (1982): Neural networks and physical systems

2.4 Van Vreeswijk & Sompolinsky (1996): Chaotic balanced state in cortical circuits

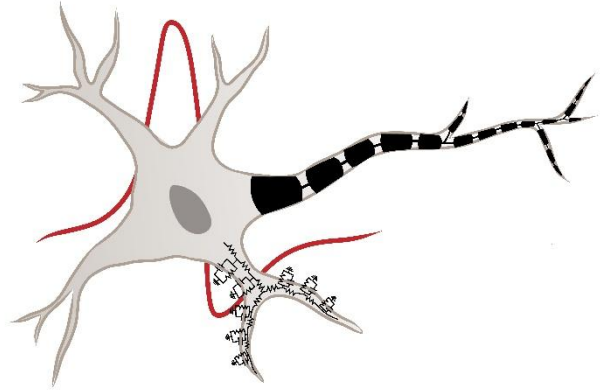
2.5 Shadlen & Newsome (1996): Motion perception, seeing and deciding

2.6 Schultz, Dayan & Montague (1997): A neural substrate of prediction and reward

2.7 Mante et al. (2013): Context-dependent computation by recurrent dynamics in prefrontal cortex

What is Computational Neuroscience?

Computational Neuroscience is an interdisciplinary field that leverages the principles and tools of anatomy, physiology, medicine, physics, biology, biochemistry, genetics, psychology, mathematics, and computer science to unravel the intricate mechanisms of the nervous system, particularly the brain. It acts as a crucial branch of neuroscience, developing and rigorously testing hypotheses about brain mechanisms through mathematical models, theoretical analyses, and abstractions of the brain. This field's scope spans multiple levels of biological organization, from molecular processes to large-scale brain networks.



A Data-driven approach for computational neuroscience, specifically focuses on transforming increasingly large-volume, high-dimensional, and multimodal experimental data into insights on the brain's structure and functions. This approach considers various levels of neuroscience, including cellular, structural, functional, medical, and behavioral aspects, and acknowledges the brain's multi-level organization from genes to the whole brain, encompassing molecules (genotypes, protein interactions), cells (morphology and electrophysiology), cellular compartments (protein localization), brain regions, the whole brain (functional and anatomical imaging), and the organism (behavior).

Hodgkin & Huxley (1952): Quantitative description of membrane current

The 1952 work by Alan Hodgkin and Andrew Huxley provided the first quantitative model of how action potentials are generated and propagated in neurons. Using voltage clamp experiments on the squid giant axon, they formulated a set of differential equations describing how voltage-gated sodium and potassium ion channels open and close in response to changes in membrane potential. Their model accurately reproduced the shape and timing of nerve impulses and laid the mathematical foundation for understanding electrical excitability in neurons. This landmark study remains a cornerstone of neurophysiology and computational neuroscience. The main equation from Hodgkin & Huxley's 1952 model describes the total membrane current per unit area as the sum of capacitive and ionic currents:

$$I_{\text{mem}} = C_m \frac{dV}{dt} + I_{Na} + I_K + I_{\text{leak}}$$

I_{mem} : the total membrane current ($\mu\text{A}/\text{cm}^2$)

$I_L = \bar{g}L(V - EL)$: the leak current

C_m : the membrane capacitance per unit area (typically $\sim 1 \mu\text{F}/\text{cm}^2$)

\bar{g} : the maximal conductance

$\frac{dV}{dt}$: the rate of change of membrane potential V over time

E : the reversal (Nernst) potentials

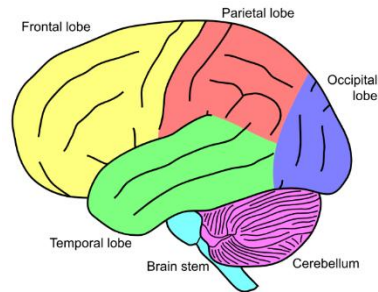
$I_{Na} = \bar{g}Na, m^3h(V - ENa)$: the sodium current

$I_K = \bar{g}K, n^4(V - EK)$: the potassium current

m, h, n : gating variables (dimensionless), which evolve over time according to voltage-dependent differential equations and describe the probability of channel states (activation and inactivation)

David Marr (1969): A theory of cerebellar cortex

In his 1969 paper, David Marr proposed a computational theory of the cerebellar cortex, suggesting that it functions as a learning system for fine-tuning motor commands through supervised learning. He theorized that the cerebellum forms associations between input patterns from mossy fibers (representing sensory and contextual information) and teaching signals from climbing fibers (representing motor error). Marr's model emphasized the role of Purkinje cells in integrating these inputs and modifying synaptic strengths to improve future responses. This work was foundational in introducing algorithmic-level explanations of brain function and helped establish the framework of understanding the brain in terms of computation, representation, and learning. Marr proposed a **basic associative learning rule** governing synaptic plasticity between parallel fibers (from granule cells) and Purkinje cells, guided by climbing fiber input:



$$\Delta w_i = \eta \cdot x_i \cdot e$$

Δw_i : the change in synaptic weight at the i^{th} parallel fiber–Purkinje cell synapse

η : learning rate constant

x_i : the activity of the i^{th} parallel fiber (granule cell)

e : the error signal provided by the climbing fiber (indicating a mismatch between intended and actual outcome)

When climbing fibers signal an error, the cerebellum adjusts the strengths of active synapses to reduce future errors (supervised learning). While Marr didn't formalize this rule explicitly with equations, this above expression captures the spirit of his computational approach.

John Hopfield (1982): Neural networks and physical systems

In his 1982 paper, John Hopfield introduced a new class of neural networks that bridged the gap between neuroscience and statistical physics. He showed that networks of symmetrically connected binary neurons could store memories as stable states or attractors in an energy landscape. When presented with partial or noisy input, the network dynamics naturally evolved toward the nearest stored pattern, demonstrating associative memory and error correction. Hopfield's formulation linked neural computation to physical systems by defining an energy function whose local minima correspond to learned patterns, laying the foundation for later developments in recurrent neural networks, attractor dynamics, and computational models of memory. The central equation in John Hopfield's 1982 paper defines the **energy function** (Lyapunov function) of the neural network. This energy function governs the network's dynamics and ensures convergence to stable states (attractors):

$$E = -\frac{1}{2} \sum_{i \neq j} w_{ij} s_i s_j + \sum_i \theta_i s_i$$

E : the total energy of the network

$s_i \in \{-1, 1\}$: the binary state of neuron i

w_{ij} : the symmetric weight between neuron i and neuron j ($w_{ij} = w_{ji}$)

θ_i : the threshold (bias) for neuron i

As the network updates neuron states asynchronously, the energy E either decreases or stays the same. This ensures that the network converges to a stable fixed point (a local energy minimum), which corresponds to a stored memory pattern. The Hopfield model thereby demonstrates associative memory through energy minimization, drawing a deep analogy between neural computation and physical systems like spin glasses.

Van Vreeswijk & Sompolinsky (1996): Chaotic balanced state in cortical circuits

In their 1996 paper, Van Vreeswijk and Sompolinsky introduced the concept of the chaotic balanced state to explain the irregular but stable firing patterns observed in cortical neurons. They modeled large networks of excitatory and inhibitory spiking neurons and showed that when excitatory and inhibitory inputs are finely balanced, the network exhibits highly irregular, asynchronous activity that is internally generated—without requiring external noise. Despite the apparent randomness, the system remains dynamically stable and responsive to stimuli. This work demonstrated that cortical variability could emerge from deterministic chaos and helped establish the balanced state as a fundamental principle of cortical network dynamics. The core of the paper is a mathematical analysis of large recurrent neural networks with excitatory and inhibitory populations, showing how a balanced state leads to chaos. A key conceptual equation represents the total synaptic input current $h_i(t)$ to neuron i :

$$h_i(t) = \sum_j J_{ij} \varphi_j(t)$$

$h_i(t)$: the total synaptic input to neuron i at time t J_{ij} : the synaptic weight from neuron j to neuron i

$\varphi_j(t)$: the output (firing rate) of neuron j at time t

They showed that for large networks with high connectivity K , excitatory (E) and inhibitory (I) contributions both grow proportionally to \sqrt{K} , but cancel each other out on average:

$$\text{Excitatory input} \sim +\sqrt{K}, \quad \text{Inhibitory input} \sim -\sqrt{K} \quad \Rightarrow \quad \text{Net input} = O(1)$$

Despite the large and opposing inputs, the net input fluctuates around zero, resulting in irregular firing and sensitivity to small perturbations (hallmarks of chaos). They also computed a positive Lyapunov exponent, confirming that the network operates in a chaotic regime, where tiny changes in initial conditions lead to diverging activity trajectories over time.

Shadlen & Newsome (1996): Motion perception, seeing and deciding

In their 1996 study, Shadlen and Newsome investigated how the brain accumulates sensory evidence to make perceptual decisions. Using a motion discrimination task in monkeys, they recorded from neurons in the lateral intraparietal area (LIP) and found that these neurons gradually increased their firing rates as the monkey viewed noisy motion stimuli. The rate and slope of this buildup correlated with both the strength of the motion signal and the animal's eventual decision. This provided compelling neural evidence for models of decision making based on temporal accumulation of evidence, such as drift-diffusion models, linking single-neuron activity directly to cognitive processes like choice and confidence. This paper did not introduce a specific new equation, but it provided strong experimental support for the use of drift-diffusion models (DDMs) in perceptual decision-making. The core equation of the drift-diffusion model, which describes how evidence is accumulated over time to reach a decision is:

$$dx = \mu dt + \sigma dW_t$$

$x(t)$: the decision variable at time t

μ : the drift rate (mean rate of evidence accumulation, based on stimulus strength)

σ : the noise amplitude

dW_t : a Wiener process (Gaussian white noise)

The decision is made when $x(t)$ crosses a threshold $\pm\theta$. In their experiments, neurons in the lateral intraparietal area (LIP) showed firing rates that ramped up over time in a way consistent with this kind of evidence accumulation. Thus, their findings provided biological evidence that neural systems might implement something mathematically equivalent to a drift-diffusion process during decision-making.

Schultz, Dayan & Montague (1997): A neural substrate of prediction and reward

In their 1997 paper, Schultz, Dayan, and Montague demonstrated that midbrain dopamine neurons encode a reward prediction error—a signal that indicates the difference between expected and received outcomes. Using recordings from behaving primates, they showed that dopamine neurons increase firing when an unexpected reward is delivered, show no change for expected rewards, and decrease firing when an expected reward is omitted. This pattern closely mirrors the error term in temporal-difference (TD) learning, a key algorithm in reinforcement learning. Their findings provided the first strong neural evidence linking dopamine to computational models of learning and decision-making, establishing a foundational connection between neuroscience and machine learning. The key equation in this paper is the temporal-difference (TD) error, which formalizes how learning occurs based on the difference between predicted and received rewards. The central equation is:

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$

δ_t : the reward prediction error at time t

r_t : the reward received at time t

$V(s_t)$: the value of the current state s_t

$V(s_{t+1})$: the value of the next state

$\gamma \in [0,1]$: the discount factor, reflecting the importance of future rewards

If the actual reward and future value are greater than expected $\Rightarrow \delta_t > 0$: dopamine neurons increase firing.

If the outcome is exactly as expected $\Rightarrow \delta_t = 0$: firing stays unchanged.

If a reward is omitted or worse than expected $\Rightarrow \delta_t < 0$: dopamine neurons decrease firing.

This TD error signal is thought to drive learning in both animals and artificial agents, adjusting expectations and guiding future behavior. The discovery that dopamine activity mirrors this error signal was a landmark in linking biological systems to reinforcement learning algorithms.

Mante et al. (2013): Context-dependent computation by recurrent dynamics in prefrontal cortex

In their 2013 study, Mante et al. investigated how the prefrontal cortex enables flexible, context-dependent decision-making. Using a task where monkeys had to selectively integrate either color or motion information depending on context, the researchers recorded neural activity and trained a recurrent neural network (RNN) to perform the same task. They found that individual neurons exhibited complex, mixed selectivity, but at the population level, the network encoded inputs, context, and decisions along distinct trajectories in neural state space. These trajectories emerged from the RNN's internal dynamics, showing that recurrent connectivity allows the brain to flexibly route relevant information depending on task demands. This study provided a mechanistic link between high-level cognition and recurrent neural dynamics. The main mathematical framework in the paper is based on a recurrent neural network (RNN) trained to perform context-dependent decision-making, mimicking prefrontal cortex behavior. The core dynamics of the RNN are governed by the standard continuous-time recurrent neural network equation:

$$\tau \frac{dx}{dt} = -x + W_{\text{rec}} \cdot \phi(x) + W_{\text{in}} \cdot u(t)$$

$x(t)$: the state vector of the recurrent units (neuronal activations) at time t

τ : a time constant governing the temporal dynamics

W_{rec} : the recurrent weight matrix

$\phi(x)$: a nonlinear activation function (sigmoid, tanh or ReLU)

W_{in} : the input weight matrix

$u(t)$: the input vector encoding stimuli and task context

The RNN is trained to integrate noisy sensory inputs selectively depending on context (motion or color). The internal dynamics (defined by W_{rec}) create low-dimensional trajectories in neural state space, explaining how the same network can flexibly route and integrate information. The trained network reproduces single-neuron and population-level dynamics observed in the monkey prefrontal cortex. This model exemplifies how recurrent dynamics can implement cognitive flexibility and contextual gating (central themes in cortical computation).