Book Notes for Theoretical Neuroscience by Dayan and Abbott

Lihao Yan

August 8, 2026

Contents

These are my personal book notes for Abbott and Dayan (2001). I am not trying to summarize the content but rather to provide high-level personal summaries, comments, and inspirations. Many important examples and concepts are skipped. Each

chapter already contains a very good summary at the end of the chapter. I am using my best judgment, but feel free to reach out about any errors or disagreements.

The textbook discusses models in theoretical neuroscience: **descriptive models** that summarize data observations, **mechanistic models** that explain the algorithms, and **interpretive models** that explain why the nervous system operates in this way. These categories echo Marr's three levels of understanding. However, complex models rarely belong to a single model type or to a single level.

1 Neural Encoding and Decoding

This section covers Abbott and Dayan (2001, § 1–4)'s content on neural encoding and decoding.

The brain interacts with the external world. The external input will be **encoded** in the neural activity, and conversely, we can **decode** the external input from the neural activity.

Definition 1.1 (Membrane Potential and Action Potential)**.** The membrane potential is the voltage difference across the cell membrane. The action potential is the rapid change in membrane potential that occurs when a neuron fires.

Remark 1. Action potentials are the only membrane-potential fluctuations that can propagate over large distances. However, neurons may communicate through other mechanisms such as glial cells, extracellular space (ephaptic effects), and volume transmission of neurotransmitters.

1.1 Neural Encoding

1.1.1 Firing Rates

The complexity, high variability, and irregularity of spike-train activity dictate the use of statistical and probabilistic descriptions. The voltage (or equivalently current) $V(t)$ is described through average firing rates (first order) and correlation functions (second order).

Abbott and Dayan (2001) describe different methods and window choices (sometimes called averaging kernels) for estimating the firing rate from experimental data.

Property 1.1 (Weber's law)**.** The just-noticeable difference (JND) in the intensity of a stimulus is proportional to the baseline intensity of that stimulus.

Property 1.2 (Fechner's law)**.** The perceived intensity of a stimulus is proportional to the logarithm of the physical stimulus intensity.

The spiking activity of neurons can be modeled as different statistical processes, such as the Poisson process. Different assumptions will lead to different variants of Poisson processes. Conversely, the Poisson process can be used to generate spiking activity.

Remark 2. The average firing rate (obtained by combining the spiking activity of the neurons, assuming independence) changes with respect to the stimulus, and therefore can encode information. Rate code focuses on information contained in the firing rate and temporal code focuses on information contained in the relative spike timings.

1.1.2 Tuning Curves

Alternative to the firing rate, $V(t)$ can be described through **tuning curves**, which transform $V(t)$ from the t-domain to the s-domain (stimulus space or physical space). The response to a stimulus is not instantaneous, so a window of the stimulus preceding a spike needs to be averaged, which gives the **spike-triggered average**.

Remark 3. The transformation is realized through different kernels $D : \mathcal{S} \mapsto \mathcal{V}$ In general, the kernel can be of arbitrary order, but the most useful and studied one is the linear response kernel.

Example 1 (Visual System)**.** The domains of the kernels of visual neurons are called receptive fields; different kernels define different types of cells—simple and complex cells with different space-time separable or non-separable receptive fields, on and off properties, and logarithmic mapping of visual space. From the encoding perspective, cells acting like kernels and cells with properties similar to those kernels (e.g. receptive field properties) have indeed been observed.

1.2 Bayesian Decoding

The decoding process can be described in a Bayesian framework where the goal is to find $P(s|r)$, the probability of the stimulus s given the firing rate r. This is closely related to the tuning curves $r(s)$.

Using standard Bayesian techniques, the optimal decoding can be obtained by minimizing loss functions such as negative likelihood loss. This probabilistic approach is closely related to information-theoretic tools such as **Fisher information**, **entropy**, and **mutual information**.

Remark 4. Probabilistic modeling works best for populations of neurons.

Remark 5. Different biological constraints imposed on the system lead to different optimized algorithms, e.g. (Calhoun, Chalasani, and Sharpee, 2014). Therefore, setting the constraints is essential.

Remark 6. Entropy rates can also be calculated on spike trains, although this is more complicated.

Example 2 (Retinal ganglion cell receptive fields)**.** See Abbott and Dayan (2001, § 2.6) for more details.

2 Neurons and Neural Circuits

This section covers Abbott and Dayan (2001, § 5–7)'s content on neuron models and neural circuits.

2.1 Single Neurons

A single neuron is modeled as a RC circuit or a spherical cow model. There are charges inside and outside the membrane, and the ions carrying the charge can pass through the membrane via a diffusion process described by a Boltzmann factor $V_{out} = V_{inside} \exp(zE/V_T)$, where z is the charge of the ion, E is the equilibrium or reversal potential, and V_T is the thermal voltage. In general, the potential takes intermediate values if multiple types of ions are present.

The single-compartment model can then be generalized to a multicompartment model that divides the neuron into compartments with distinct potential dynamics.

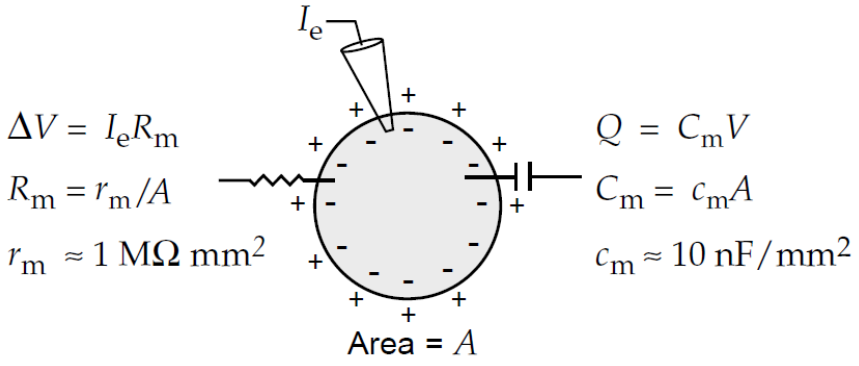

Figure 1: Single-compartment neuron as a membrane resistor–capacitor circuit ((Abbott and Dayan, 2001, Fig. 5.3)).

Remark 7. This picture could be further refined by including more dependence on how each component (e.g. conductance, more V compartments, capacitance, etc.) is changing. This already has the core ingredients for all components that affect the electric fields, and therefore the membrane current.

Definition 2.1 (Membrane Current)**.** The membrane current is the current flowing through the membrane across all ion channels.

Example 3 (Integrate-and-fire model and the Hodgkin–Huxley model)**. Integrate-and-fire model** does not model action potentials explicitly, but artificially stipulates an action potential whenever the membrane potential $V(t)$ reaches a threshold value $V_{\text{threshold}}$. Therefore, the integrate-and-fire model focuses on the dynamics of the subthreshold membrane potential. The model is described by the following differential equation:

$$\tau_m \frac{dV}{dt} = E_L - V(t) + R_m I(t), \tag{1}$$

where E_L is the resting potential, R_m is the membrane resistance, and $I(t)$ is the input current. The firing rate is defined through the interspike-interval calculation. Fancier mechanisms such as spike-rate adaptation and refractoriness can be incorporated by changing the dynamics of the conductances, which can be grounded in the biophysical properties of the ion channels. For example, the current conductance equals the maximum conductance times the channel open probability.
Hodgkin–Huxley model is a more biophysical model that models the action potential explicitly by modeling the conductances through dimensionless gating variables.

Example 4 (Synaptic Conductance)**.** Similar to modeling membrane conductances, synaptic conductances governing synaptic transmission can be modeled through the release probability of the transmitters and the receptor binding probability. The synaptic conductance can also be included in the leaky integrate-and-fire model.

Remark 8. This model can capture both regular and irregular firing modes through different balances of excitatory and inhibitory inputs.

Remark 9. How do receptive-field properties and other properties arise at the single-cell level? Most likely this is through network connectivity, but this can also arise through dendritic computation (Poirazi and Papoutsi, 2020; London and Häusser, 2005).

Remark 10. Attenuation of neural electrical activity along the long, narrow cable-like structures (dendritic or axonal branches) can be described through cable theory or Maxwell's equations.

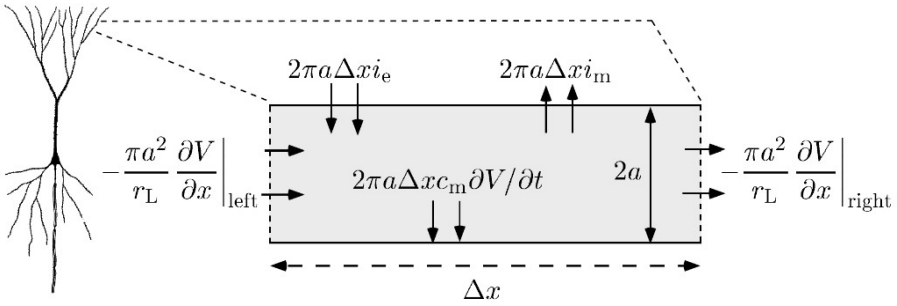

Figure 2: Segment of a neuron for the derivation of the cable equation ((Abbott and Dayan, 2001, Fig. 6.6)).

Example 5 (Myelinated axons)**.** The capacitance of a myelinated axon can be approximated by a model that treats the membrane as an extremely thick cell membrane.

2.2 Network Models

Spiking models are useful, but they have many more free parameters to analyze and are harder to simulate. Abbott and Dayan (2001, § 7) cover firing-rate models, which are simpler and more tractable. This line of research is closely related to modern artificial neural network and results in modern neural network-based artificial intelligence.

Definition 2.2 (Firing Rate Model)**.** A firing-rate model describes the firing rate of a neuron as a function of the input current

$$r(t) = f(I(t)), \tag{2}$$

where $r(t)$ is the firing rate and $I(t)$ is the input current. The input current is described by a differential equation similar to that in the section above.

Firing-rate models and spiking models capture different aspects of neural activity. A firing-rate model is useful when neurons in a network are uncorrelated, with little synchronous firing, and when precise spike timing is unimportant. Firing-rate models are also easier to simulate and analyze.

Example 6 (Feedforward and recurrent networks)**.** The RNN is given by

$$\tau_r \frac{d\mathbf{v}}{dt} = -\mathbf{v} + F(\mathbf{W} \cdot \mathbf{u} + \mathbf{M}\mathbf{v}) + h, \tag{3}$$

where $\mathbf{v}$ is the vector of firing rates, $\mathbf{u}$ is the vector of inputs, F is a nonlinear function, $\mathbf{W}$ is the weight matrix of the feedforward connections, and $\mathbf{M}$ is the weight matrix of the recurrent connections. h is an external input. The network is purely feedforward if $\mathbf{M}$ is set to zero.

Property 2.1 (Dale's principle)**.** Dale's principle states that a neuron releases the same set of transmitters at all of its synapses; whether a target is excited or inhibited depends on the postsynaptic receptors, not on the identity of the presynaptic cell alone.

Abbott and Dayan (2001, § 7) build these networks by manually fixing the weights. These networks can achieve input integration, coordinate transformation, amplification, modulation, and associative memory (Hopfield, 1982).

Example 7 (Excitatory–inhibitory networks)**.** They use stability analysis to show that a network with excitatory and inhibitory neurons can lead to different types of dynamical behavior, such as a Hopf bifurcation.

Example 8 (Boltzmann machine)**.** A Boltzmann machine is a type of stochastic network in which the input–output relationship is stochastic. In the simplest case, the unit takes values 0 and 1 with probabilities $1 - p$ and p, respectively. The probability is given by the total input current:

$$p = \frac{1}{1 + e^{-I_a}}, \tag{4}$$

where I_a is the total input current defined as

$$I_a(t) = h_a(t) + M\mathbf{v}(t), \tag{5}$$

where $h_a(t)$ is the total feedforward input to unit a, and M is a symmetric weight matrix of the recurrent connections. The network evolves as a **Markov chain**; this update procedure is called **Gibbs sampling**, and its dynamics are known as **Glauber dynamics**. The network follows a Boltzmann distribution when it converges to an equilibrium state.

3 Adaptation and Learning

This section covers Abbott and Dayan (2001, § 8–10)'s content on adaptation and learning. Activity-dependent synaptic plasticity is widely believed to be the basic phenomenon underlying learning, memory, and development.

3.1 Hebbian Learning

Property 3.1 (Hebb rule)**.** The Hebb rule postulates that synapses change in proportion to the correlation or covariance of the activities of the pre- and postsynaptic neurons.

The Hebb rule can be introduced into a firing-rate model (3) as

$$\tau_w \frac{d\mathbf{W}}{dt} = g(\mathbf{v}, \mathbf{u}), \tag{6}$$

where $\mathbf{u}$ is the presynaptic activity and $\mathbf{v}$ is the postsynaptic activity. g is a general function of the presynaptic and postsynaptic activities, which can lead to different dynamics of the weight matrix. In general, g can involve global information (e.g. synaptic normalization) or activity at different t (e.g. timing-based rules).

Remark 11. A Hebbian rule with a suitable g can converge to an optimal solution for the relevant loss function (e.g. PCA's reconstruction error). g dictates the learning dynamics.

Example 9 (Applications of Hebbian learning)**.** By choosing g properly, Hebbian learning can be used to develop *ocular dominance*, *orientation selectivity*, and *invariant responses*. When using a network with recurrent connections, Hebbian learning rules can be used to develop ocular dominance stripes. Note that constraints are important for Hebbian learning to converge to the desired solution.

With realistic constraints, the learning rule can be generalized (e.g. the elastic net rule and anti-Hebbian modification). Hebbian learning can also generalize to a supervised learning setting by providing both input $\mathbf{u}$ and output $\mathbf{v}$. Further learning generalizations are **gradient descent** and the **delta rule**, which are important in modern machine learning.

Additionally, they present contrastive Hebbian learning. Similar contrastive learning algorithms have been used to train modern artificial neural networks. The probabilistic learning scheme, which uses a neural network to learn distributions, is related to generative modeling in modern machine learning.

Remark 12. One benefit of Hebbian learning is that it is a local learning rule and is much more biologically plausible than modern gradient descent-based learning rules.

Remark 13 (Hebbian learning and backpropagation). Modern machine learning uses backpropagation to train neural networks. It has been shown that Hebbian learning could be used to approximate backpropagation signals (Lillicrap et al., 2020).

3.2 Classical Conditioning and Reinforcement Learning

In Abbott and Dayan (2001, § 9), they cover classical conditioning and reinforcement learning. They discuss temporal-difference learning and immediate reward versus delayed reward.

Temporal-difference learning is a version of the delta rule we discussed earlier, but it uses the total expected future reward and has specific policies for choosing actions. This is related to **Q-learning** in modern reinforcement learning (Sutton and Barto, 2020), where Q is the future reward information.

Example 10 (Actor–critic method)**.** Methods that learn approximations to both the policy and the value function are often called actor–critic methods (Sutton and Barto, 2020).

3.3 Representational Learning

Representational learning is similar to modern **generative modeling**, where the core idea is to represent the hidden structure (causes) by analyzing the statistical structure of visual images. The images are represented by different distributions $p(\mathbf{x})$.

The expectation-maximization algorithm is used to find the $p(\mathbf{x})$ that best matches the real data. The essence is to minimize loss functions such as likelihood loss.

Generative modeling is widespread in both image generation and language generation. One interesting line of work related to invertible probabilistic models is **normalizing flows** (Kobyzev, Prince, and Brubaker, 2021), where flow means the chain of transformations of the probability distributions.

In the text, they discuss the following models: Gaussian mixture models, factor analysis models, and sparse coding models. Let u be the data and v be the latent variables (causes). The prior distribution is $p_\theta(v)$ and the generative model (likelihood) is $p_\theta(u|v)$. These models use different priors and generative models. The ultimate goal is to find the optimal θ that maximizes the likelihood $p_\theta(u)$, i.e. $p_\theta(u) \simeq p_{\text{data}}(u)$. $p_\theta(u)$ is given by Bayes' rule:

$$p_\theta(u) = \int p_\theta(u|v)p_\theta(v)\mathrm{d}v. \tag{7}$$

Example 11 (K-means, principal component analysis, and independent component analysis)**.** K-means is a limiting case of a mixture of Gaussians, principal component analysis is a limiting case of factor analysis, and independent component analysis is a limiting case of sparse coding, all obtained by taking the noise variances to zero (yielding deterministic recognition).

Remark 14. Many interesting discussions in Abbott and Dayan (2001, § 10) were skipped. However, the statistical learning literature have been greatly developed in recent years.

References

Abbott, Laurence F. and Peter Dayan (2001). Theoretical Neuroscience. Cambrdige, MA: MIT Press. URL: https://mitpress.mit.edu/9780262041997/theoretical-neuroscience/ (visited on 09/06/2023).

Calhoun, Adam J, Sreekanth H Chalasani, and Tatyana O Sharpee (Dec. 9, 2014). "Maximally Informative Foraging by Caenorhabditis Elegans". In: eLife 3. Ed. by Ranulfo Romo, e04220. ISSN: 2050-084X. DOI: 10.7554/eLife.04220. URL: https://doi.org/10.7554/eLife.04220 (visited on 07/24/2026).

Hopfield, J. J. (Apr. 1982). "Neural Networks and Physical Systems with Emergent Collective Computational Abilities." In: Proceedings of the National Academy of Sciences 79.8, pp. 2554–2558. DOI: 10.1073/pnas.79.8.2554. URL: https://www.pnas.org/doi/10.1073/pnas.79.8.2554 (visited on 12/14/2022).

Kobyzev, Ivan, Simon J. D. Prince, and Marcus A. Brubaker (Nov. 1, 2021). "Normalizing Flows: An Introduction and Review of Current Methods". In: IEEE Transactions on Pattern Analysis a 43.11, pp. 3964–3979. ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2020.2992934. arXiv: 1908.09257 [cs, stat]. URL: http://arxiv.org/abs/1908.09257 (visited on 02/26/2023).

Lillicrap, Timothy P. et al. (June 2020). "Backpropagation and the Brain". In: Nature Reviews Neuroscience 21.6, pp. 335–346. ISSN: 1471-0048. DOI: 10.1038/s41583-020-0277-3. URL: https://www.nature.com/articles/s41583-020-0277-3 (visited on 08/06/2026).

London, Michael and Michael Häusser (July 21, 2005). "DENDRITIC COMPUTATION". In: Annual Review of Neuroscience 28 (Volume 28, 2005), pp. 503–532. ISSN: 0147-006X, 1545-4126. DOI: 10.1146/annurev.neuro.28.061604.135703. URL: https://www.annualreviews.org/content/journals/10.1146/annurev.neuro.28.061604.135703 (visited on 07/24/2026).

Poirazi, Panayiota and Athanasia Papoutsi (June 2020). "Illuminating Dendritic Function with Computational Models". In: Nature Reviews Neuroscience 21.6, pp. 303–321. ISSN: 1471-0048. DOI: 10.1038/s41583-020-0301-7. URL: https://www.nature.com/articles/s41583-020-0301-7 (visited on 07/24/2026).

Sutton, Richard S. and Andrew Barto (2020). Reinforcement Learning: An Introduction. Second edition. Adaptive Computation and Machine Learning. Cambridge, Massachusetts London, England: The MIT Press. 526 pp. ISBN: 978-0-262-03924-6.