Recitation 0

Course Introduction & Lecture Review

Qilin Zhang, GPT6 astra

Computational Neuroscience, Fall 2026

September 27, 2026

Today

Course Policies

- Course Resources
- Homework
- Final Project
- Conclusion

Review of Previous Lectures

- Neurons, Synapses, and Circuits
- Wiring and Dendrites
- Perceptrons and Representation Geometry

Part I

Course Policies

Resources, Homework and Final Project

Course Resources

We have three platforms for the course:

- **QQ**: For daily chat, material release, announcements, and textbooks.
- **USTC BB**: Homework submission.
- **Course website**[1]: You can find latest course materials, introduction to policies and time table. Use the discussion issues page[2] to make discussions and get bonus points.

[1] https://wenlab.github.io/Computation-Neuro-Course/

[2] https://github.com/Wenlab/Computation-Neuro-Course/issues

Homework

Each time the homework is released by the form of a problem set. Homework constitutes about 70% of your final grade.

- **Submission**: The homework should be submitted on USTC BB before the deadline.
- **Deadline**: In this course, deadline is dead line. If not inform TA in advance, late submission will be penalized by 2 points per day. Please email[3] the TA if you have any special circumstances.
- **Collaboration**: Please state your collaborators in your homework, including your AI tools.

[3] zhangqilin8879@mail.ustc.edu.cn

Final Project

The final project constitutes about 30% of your final grade.

A preliminary plan for the final project is a group project for 3-4 students.

Conclusion: How to get higher grades?

The final score will be calculated as follows:

$$\text{Final Score} = 0.7 \times \text{Homework} + 0.3 \times \text{Final Project} + \text{Bonus Points}.$$

According to experience, the following suggestions may be very helpful:

- Submit homework on time.
- Post and answer questions on the issue page. Both will give you bonus points.

It is unavoidable to use AI tools today, but we still need to learn. Please make sure that you have understood your work before submission.

Part II

Review of Previous Lectures

Chapter 1: Connecting Neurons to Perform Computation

Review Overview

Two recurring questions in computational neuroscience are:

- How do neural structures support computation?
- How can we understand the computation in neural systems?

To answer these questions, we need to use models to connect theory and experiment.

A good model need to be **analyzable** and **biologically plausible**.

- Analyzable means we can provide qualitative and quantitative results.
- Biologically plausible means the mechanism could be realized by neural circuits.

In Chapter 1, we chose three topics to show how researchers consider these questions and build models.

Review Overview: Why we need these topics in Chapter 1?

There are three main topics in Chapter 1,

- Neurons and synapses: the basic motifs and units in neural circuits.
- Wiring optimization: why neurons need their compartments.
- Perceptrons: some theoretical properties of a simple computation model.

The first topic provides an introduction to the basic computation in neural circuits, and it provides a foundation of **biological plausibility** for models in subsequent topics.

The second topic is about how to interpret the morphology features in neural circuits in a manner of **optimization**, i.e., to better support computation.

The third topic is a simple model which can help us understand the computation in a feedforward array of neurons. It shows the potential of a model with brevity, and builds a connection to learning theory and **modern machine learning**.

Topic 1

Neurons, Synapses, and Circuits

Modeling, Abstraction and Gaps

From Membrane Physics to a Neuron Model

- Ion gradients and selective permeability establish reversal potentials.
- The membrane stores charge; ionic conductances drive its voltage.

$$C_m \frac{dV}{dt} = -\sum_i g_i (V - E_i) + I_{\text{ext}}.$$

A useful simplification

The leaky integrate-and-fire model retains leakage, temporal integration, and threshold crossing, with a reset after each spike.

Under constant input, the charging time determines the firing rate, and a dynamical model becomes an **input–output relation.**

Synapses Change Conductance

Synaptic effects depend on both conductance and driving force:

$$I_{\text{syn}} = g_{\text{syn}}(t)(V - E_{\text{syn}}).$$

If the inhibitory reversal potential equals E_L, the steady voltage is

$$V - E_L = \frac{g_{\text{exc}}}{g_L + g_{\text{inh}} + g_{\text{exc}}}(E_{\text{exc}} - E_L).$$

- Shunting inhibition changes the response to excitation through the **total conductance** in the denominator.
- A divisive effect on voltage does not imply purely divisive firing-rate modulation under every input condition.

From Synaptic Interactions to Computation

Example: motion direction selectivity

Spatially arranged inputs with different time courses interact through excitation and inhibition, producing different responses to opposite directions of motion.

- Convergence and divergence organize the collection and distribution of signals.
- Feedforward, feedback, and lateral interactions organize how excitation and inhibition act on those signals.
- Some small connectivity motifs occur more often than expected under the random-network comparison used in the lecture.

Connectivity Does Not Determine Function Alone

- Cerebral and cerebellar circuits illustrate different arrangements of convergence, divergence, and neuronal populations.
- Connectivity must be interpreted together with intrinsic neuronal dynamics and synaptic properties.
- In pyloric circuit models, substantially different parameter combinations can produce similar activity patterns.

What does a successful model establish?

Reproducing an activity pattern demonstrates a possible mechanism; it does not uniquely identify the underlying parameters.

Lecture example: Prinz, Bucher & Marder (2004).

Summary: Abstraction and Gaps based on Modeling

In this topic, we need to remember these points:

- Neurons perform computation by potential. Synapses tune the connection between neurons.
- The electrical properties of neurons can be modeled by dynamical systems, but we use **rate model for simplicity**.
- Neurons wire together to do computation. Both the wiring and the mechanics are important.

We can find that a key difference between computational neuroscience and machine learning is that **we cannot directly observe all the parameters**.

$$\text{Ion channels} \rightarrow \text{Single-neuron dynamics} \rightarrow \text{Circuit motifs}$$

Some discussions

The inverse problem

Different parameter combinations can produce similar neural activity. A successful fit is therefore a possible mechanism rather the unique solution.

$$\mathcal{M}_y = \{\theta : I(\theta) = y\}$$

- The set $\mathcal{M}_y$ contains all model parameters compatible with the chosen observations.
- Its directions describe compensations: one parameter can change while another preserves the same function.
- The authors use diffusion models to sample and study this set.

Ruilin Zhang, Louis Tao, and Zhuo-Cheng Xiao, "Diffusion learning reveals viable parameter manifolds and compensation geometry in biological dynamical systems," arXiv:2607.03671v2 (2026).

Topic 2

Wiring and Dendrites

Neuron Morphology in an Optimization Perspective

Why Axons, Dendrites, and Spines?

How much space is needed to satisfy the same potential connectivity?

Wiring design	Network size R
Point-to-point axons	Nd
Branching axons	$N^{5/6}d$
Axons and dendrites	$N^{2/3}d$
Adding dendritic spines	$N^{2/3}d^{4/3}/s^{1/3}$

Scaling estimates: N neurons, process diameter d, spine reach s.

Branching, dendrites, and spines progressively reduce the space needed to meet the model's connectivity requirement.

What Does Wiring Optimization Explain?

- **Requirement:** processes must be close enough to permit a synapse. Potential connectivity is not actual all-to-all connectivity.
- **Assumptions:** simplified spatial packing, fixed process diameters and spine reach, and negligible spine volume in the estimate.
- **Prediction:** the symmetric model gives a process-length scale $l \sim Nd^2/s$; allowing different axonal and dendritic diameters supports comparison with anatomical lengths.

Interpretation

Wiring economy offers a quantitative explanation for morphological features under stated constraints, rather than a universal proof that every neuron must have them.

From a Point Neuron to a Cable

Long, narrow processes can have spatially varying membrane potentials. For a uniform passive cable, away from input sites,

$$\tau_m \frac{\partial v}{\partial t} = \lambda^2 \frac{\partial^2 v}{\partial x^2} - v, \qquad v = V - E_L.$$

$$\tau_m = c_m r_m, \qquad \lambda = \sqrt{\frac{a r_m}{2\rho_L}}.$$

- τ_m: membrane time scale; λ: spatial decay scale.
- Constant point input on an infinite cable: $v(x) \propto e^{-|x|/\lambda}$.
- An impulse spreads with variance $2\lambda^2 t/\tau_m$, while leakage reduces its spatial integral as e^{-t/τ_m}.

How Can Signals Travel Faster?

- **Increase axonal radius:** the passive-cable peak-delay estimate scales as $s_{\text{eff}} \sim \lambda/\tau_m \propto \sqrt{a}$ when membrane properties are fixed.
- **Add myelin:** insulating layers reduce effective capacitance and increase membrane resistance, improving signal transmission.
- **Optimize geometry:** at fixed outer radius, the idealized sheath model predicts

$$\frac{a_{\text{inner}}}{a_{\text{outer}}} = e^{-1/2} \approx 0.61.$$

These are simplified scaling arguments. The passive response is not a complete model of regenerative action-potential propagation.

Dendrites Beyond Passive Cables

- Backpropagating action potentials can interact with distal input to trigger dendritic calcium spikes.
- Active sensing is associated with specific dendritic calcium signals in the lecture's experimental example.
- Local nonlinear integration motivates treating dendritic branches as computational subunits.

A bridge to abstract models

A linear threshold unit is a useful abstraction, but it can omit important computations within a single neuron.

Lecture examples: Larkum et al. (1999); Xu et al. (2012); Poirazi, Brannon & Mel (2003).

Topic 3

Perceptrons and Representation Geometry

From classical theory to modern feature learning

From a Neuron to a Classifier

A linear threshold unit turns an input vector into a binary decision:

$$y = \text{sign}(\mathbf{w}^\top \mathbf{x} - \theta), \qquad y \in \{-1, +1\}.$$

- $\mathbf{x}$: input activities or features; $\mathbf{w}$: weights; θ: decision threshold.
- The abstraction retains weighted integration and a threshold, while omitting membrane dynamics and dendritic structure.
- McCulloch–Pitts units emphasize logical computation; perceptron learning adds an error-driven weight adjustment.

Two distinct questions

Which classifications can this model implement?
How can it learn weights for a particular classification?

Cover's Theorem: What Are We Counting?

Fix P input vectors in $\mathbb{R}^N$ and vary $\mathbf{w}$ in $y = \text{sign}(\mathbf{w}^\top \mathbf{x})$.

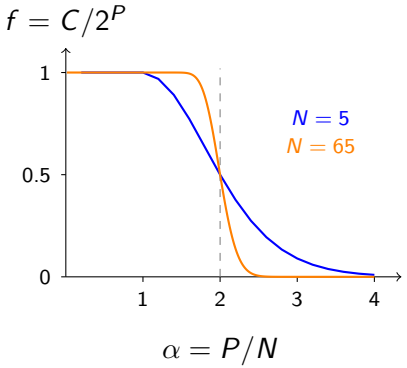
- **General position:** every subset of at most N input vectors is linearly independent.
- **Unrestricted real weights; zero threshold.** Inputs on the decision boundary do not count as strictly separated.

$$\boxed{C(P,N) = 2\sum_{k=0}^{N-1} \binom{P-1}{k}}, \qquad f(P,N) = \frac{C(P,N)}{2^P}.$$

C counts realizable binary labelings; f is the probability that a uniformly random labeling is separable.

A trainable threshold can be absorbed into an extra input coordinate: $[\mathbf{w}, \theta]^\top [\mathbf{x}, -1]$. The effective dimension then increases by one.

Capacity as a Transition

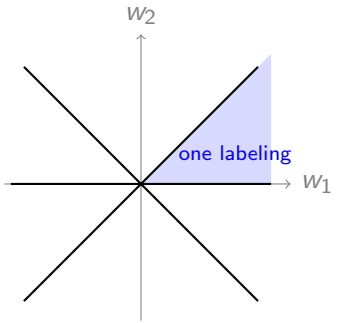


- $P \leq N$: all 2^P labelings are realizable.
- $P = 2N$: exactly half are realizable.
- At fixed α, as $N \to \infty$, $f \to 1$ for $\alpha < 2$ and $f \to 0$ for $\alpha > 2$.

How to interpret $\alpha_c = 2$

A threshold for random-label separability, not a guarantee for every labeling below $2N$, and not a prediction of test accuracy.

The Same Problem in Weight Space



Three generic lines: $C(3,2) = 6$ regions.

Each input defines a boundary in weight space:

$$H_\mu = \{\mathbf{w} : \mathbf{w}^\top \mathbf{x}^\mu = 0\}.$$

A prescribed label selects one half-space:

$$y^\mu \mathbf{w}^\top \mathbf{x}^\mu > 0.$$

- A solution must satisfy every constraint.
- Each region corresponds to one realizable labeling.
- Adding a boundary splits some existing regions.

$$C(P+1, N) = C(P, N) + C(P, N-1).$$

From Fixed Representations to Feature Learning

Classical question

Fix the representation $\{\mathbf{x}^\mu\}$ and ask whether a linear readout can separate the labels. Cover's theorem and weight-space geometry quantify this capacity.

Modern extension

Training can reshape the representation itself: task-related activity manifolds become easier to separate. **Manifold capacity** measures this geometric change.

$$\text{Fixed representation} \longrightarrow \text{learned representation}$$

Chi-Ning Chou, Hang Le, Yichen Wang, and SueYeon Chung, "Feature Learning beyond the Lazy-Rich Dichotomy: Insights from Representational Geometry," ICML 2025, arXiv:2503.18114v2.

Learning by Correcting Mistakes

Present labeled examples $(\mathbf{x}^\mu, y^\mu)$ repeatedly. With a fixed learning rate $\eta > 0$,

$$\mathbf{w} \leftarrow \begin{cases} \mathbf{w} + \eta y^\mu \mathbf{x}^\mu, & y^\mu \mathbf{w}^\top \mathbf{x}^\mu \le 0, \\ \mathbf{w}, & y^\mu \mathbf{w}^\top \mathbf{x}^\mu > 0. \end{cases}$$

- Update only when an example is wrong or on the boundary.
- On that example, the signed score increases by $\eta \|\mathbf{x}^\mu\|^2$.
- Other examples may become incorrect: progress is not a monotonic decrease in the total number of errors.

Stop after a complete pass with no updates.

Why Does the Algorithm Converge?

Assume a finite, strictly separable dataset, $\mathbf{w}_0 = 0$, and repeated presentation of every example. For a separating $\mathbf{w}^*$,

$$\delta = \min_{\mu} \frac{y^{\mu} \mathbf{w}^{*\top} \mathbf{x}^{\mu}}{\|\mathbf{w}^*\|} > 0, \qquad D = \max_{\mu} \|\mathbf{x}^{\mu}\|.$$

- Each update increases alignment with $\mathbf{w}^*$ by at least $\eta\delta\|\mathbf{w}^*\|$.
- After M updates, $\|\mathbf{w}_M\| \leq \eta D\sqrt{M}$. Alignment cannot exceed the product of the two norms.

$$\boxed{M \leq \left(\frac{D}{\delta}\right)^2}.$$

Interpretation: a larger separating margin gives a tighter update bound. For nonseparable data, this convergence guarantee does not apply.

From Learning Algorithm to Modern Problems

Delta Rule $\Longrightarrow$ kernel regression

For fixed neural responses $\mathbf{r}(\theta)$ and squared error, the Delta Rule converges to a least-squares readout. Its prediction can be written as

$$f(\theta) = \sum_{\mu} \alpha_{\mu} K(\theta^{\mu}, \theta), \qquad K(\theta, \theta') = \frac{1}{N} \mathbf{r}(\theta)^{\top} \mathbf{r}(\theta').$$

For binary labels, the classifier is $\operatorname{sign} f(\theta)$.

Modern kernel-learning view

The kernel spectrum controls how the readout generalizes. When model complexity or sample size crosses the interpolation threshold, test error can first rise and then fall again: the **double-descent** curve. Thus convergence finds a training solution, but does not by itself determine out-of-sample performance.

Bordelon & Pehlevan, 2021; kernel-regression formulation in the cited work.

What Does the Perceptron Teach Us?

Capacity: does a solution exist?

Cover's theorem counts realizable labelings under a specified geometric assumption. It does not prescribe how weights are found.

Learning: can an algorithm find a solution?

The perceptron algorithm finds a separator for strictly separable data; the margin controls its worst-case update bound.

The abstraction makes two aspects of learning explicit: **the geometry of representations** and **the dynamics of weight adjustment**.

Neither result alone guarantees good performance on unseen inputs.

Recitation 0

Qilin Zhang, GPT6 astra

Course Policies

Lecture Review

Neurons and Circuits

Wiring and Dendrites

Perceptrons

Questions and Discussion

Questions?