Low-Rank Recurrent Networks: From Connectivity to Collective Computation

October 19, 2026

Roadmap. Lecture 6 explained how random connectivity generates self-consistent fluctuations. Here we ask how a small amount of structured connectivity organizes a network into an interpretable dynamical system. The main calculation reduces a rank-one network to one collective variable. We use it to explain persistent memory and input selection, and finish with a brief discussion of random connectivity and context-dependent gain. The core material is intended for approximately 25–30 minutes; the final population-structure example can be treated qualitatively.

Notation. We retain the current-based convention of Lecture 6, with time measured in units of τ_r:

$$\dot{\mathbf{x}} = -\mathbf{x} + \mathbf{J}\phi(\mathbf{x}) + \mathbf{h}(t), \qquad r_i = \phi(x_i). \tag{1}$$

Here J_{ij} connects neuron j to neuron i, and $\phi(x) = \tanh x$ describes signed output. The framework follows Mastrogiuseppe and Ostojic [1]; the final example draws on Dubreuil et al. [2].

Adding a small number of organized feedback pathways

Write the connectivity as

$$\boxed{\mathbf{J} = g\boldsymbol{\chi} + \mathbf{P}, \qquad \mathbf{P} = \frac{1}{N}\sum_{a=1}^{R} \mathbf{m}^{(a)}(\mathbf{n}^{(a)})^T, \qquad R \ll N.} \tag{2}$$

The entries of $\boldsymbol{\chi}$ are independent centered Gaussians with variance $1/N$. The structured vectors are fixed during the dynamics and independent of $\boldsymbol{\chi}$. Their components and empirical second moments remain of order one as N grows; R is fixed. The random part is usually full rank, so it is $\mathbf{P}$, not generally $\mathbf{J}$, that has low rank. Low rank also does not imply sparsity: P_{ij} can be nonzero for every pair of neurons.

For rank one,

$$P_{ij} = \frac{m_i n_j}{N}, \qquad \mathbf{P}\mathbf{r} = \mathbf{m}\, a, \qquad a = \frac{\mathbf{n}^T \mathbf{r}}{N}. \tag{3}$$

Every recurrent output is a multiple of the same vector $\mathbf{m}$. The vector $\mathbf{n}$ first reads a weighted population signal a; the vector $\mathbf{m}$ distributes that scalar back to the neurons. This is a feedback loop through a single collective channel. Reading and feedback need not use the same weights, so $\mathbf{P}$ need not be symmetric.

Why the structured weights scale as $1/N$

In Lecture 6, independent random weights scaled as $N^{-1/2}$ so that a sum of incoherent contributions had finite variance. Here the structured input is

$$\sum_j P_{ij} r_j = m_i \left(\frac{1}{N} \sum_j n_j r_j \right).$$

If the output has a nonzero order-one overlap with $\mathbf{n}$, the sum in parentheses remains finite. Coherent feedback therefore survives with individual weights of order N^{-1}. If n_j and r_j are instead centered and effectively uncorrelated, that overlap is typically only $O(N^{-1/2})$. The structured effect depends on alignment, not just on the size of a single synapse. Even when each structured weight is smaller than a typical random weight, its collective effect need not be small.

The exact rank-one reduction

To isolate the geometry, first set $g = 0$ and $\mathbf{h} = 0$. Then

$$\dot{\mathbf{x}} = -\mathbf{x} + \mathbf{m}\, a(t), \qquad a(t) = \frac{1}{N} \sum_i n_i \phi(x_i(t)). \tag{4}$$

Assume $\mathbf{m} \neq 0$. Decompose the current vector into its component along $\mathbf{m}$ and an orthogonal residual:

$$\mathbf{x} = \mathbf{m}\kappa + \mathbf{x}_\perp, \qquad \kappa = \frac{\mathbf{m}^T \mathbf{x}}{\mathbf{m}^T \mathbf{m}}, \qquad \mathbf{m}^T \mathbf{x}_\perp = 0. \tag{5}$$

Projection of Equation 4 gives

$$\dot{\kappa} = -\kappa + a(t), \qquad \dot{\mathbf{x}}_\perp = -\mathbf{x}_\perp. \tag{6}$$

Consequently $\mathbf{x}_\perp(t) = e^{-t} \mathbf{x}_\perp(0)$. The recurrent nonlinearity cannot regenerate an orthogonal current component because its output is always parallel to $\mathbf{m}$. On the invariant subspace $\mathbf{x} = \mathbf{m}\kappa$, the closed equation is

$$\boxed{\dot{\kappa} = -\kappa + F(\kappa), \qquad F(\kappa) = \frac{1}{N} \sum_i n_i \phi(m_i \kappa).} \tag{7}$$

This equation is exact at finite N for initial conditions in that subspace. For general initial conditions, the additional term $\mathbf{x}_\perp(0)e^{-t}$ remains inside ϕ during the transient. No Gaussian approximation or DMFT is needed for this reduction.

A distinction worth keeping. The coordinate κ and the rate overlap a are different quantities. They obey $\dot{\kappa} = -\kappa + a$, and coincide at a fixed point, not generally during a trajectory. We use separate symbols because both are useful when random connectivity is added.

Heterogeneous neurons, coordinated activity

For $\mathbf{x} = \mathbf{m}\kappa$, neuron i has current $m_i\kappa$ and output $\phi(m_i\kappa)$. Different coefficients produce different amplitudes, signs, and degrees of saturation, although one variable determines the population state. A low-dimensional collective description is therefore compatible with heterogeneous single-neuron responses.

The current vector lies in a one-dimensional linear subspace. The output vector $\boldsymbol{\phi}(\mathbf{m}\kappa)$ generally traces a curved one-dimensional manifold; it need not lie in a one-dimensional linear subspace. Thus several principal components can be needed to describe the outputs even when the intrinsic dynamical dimension is one. Low connectivity rank, intrinsic state dimension, and PCA dimension are related but distinct.

Feedback overlap and linear stability

Assume $\phi(0) = 0$. Linearizing Equation 7 gives

$$\dot{\kappa} = [-1 + \phi'(0)\lambda]\,\kappa + O(\kappa^2), \qquad \lambda = \frac{\mathbf{n}^T\mathbf{m}}{N}. \tag{8}$$

For an odd smooth nonlinearity the next term is cubic. The same overlap is the potentially nonzero eigenvalue of the structured connectivity:

$$\mathbf{P}\mathbf{m} = \lambda\mathbf{m}, \qquad \mathbf{n}^T\mathbf{P} = \lambda\mathbf{n}^T.$$

All remaining eigenvalues are zero. If $\lambda = 0$, a nonzero rank-one matrix can still exist; it is then nilpotent rather than possessing a nonzero eigenvalue.

For tanh, $\phi'(0) = 1$, and the origin is locally stable for $\lambda < 1$. Just below one, the collective decay time is

$$\tau_{\text{eff}} = \frac{\tau_r}{1 - \lambda}. \tag{9}$$

This produces slow decay and approximate integration over a finite time. At $\lambda = 1$ the linear decay vanishes, but nonlinear terms generally remain: marginal linear stability is not an exact nonlinear line attractor.

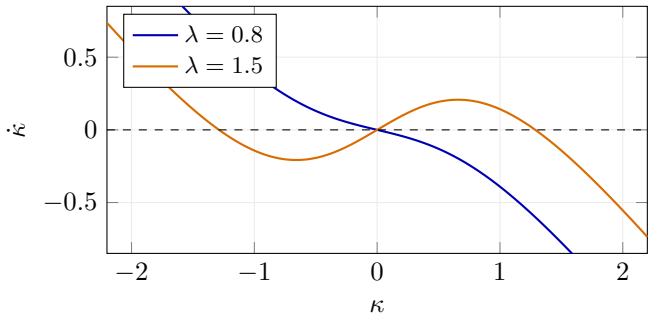


Figure 1: The exact scalar vector field for the binary-pattern example. A positive value moves κ rightward and a negative value moves it leftward. Increasing feedback through $\lambda = 1$ destabilizes zero and creates two stable nonzero states. These curves are analytic vector fields, not network simulations.

A worked example: persistent binary memory

Choose a pattern with $m_i \in \{-1, +1\}$ and set

$$n_i = \lambda m_i, \qquad \lambda > 0. \tag{10}$$

Since $m_i \tanh(m_i \kappa) = \tanh \kappa$, the reduced equation is exactly

$$\boxed{\dot{\kappa} = -\kappa + \lambda \tanh \kappa.} \tag{11}$$

The fixed points solve $\kappa_* = \lambda \tanh \kappa_*$. For $\lambda < 1$, zero is the only fixed point and is stable. For $\lambda > 1$, zero is unstable and two stable states appear at $\pm\kappa_*$. At a nonzero fixed point, the scalar Jacobian is

$$f'(\kappa_*) = -1 + \lambda \operatorname{sech}^2 \kappa_* < 0.$$

The orthogonal current components also decay, so these are stable states of the full purely rank-one network.

Near the transition, expand $\tanh \kappa = \kappa - \kappa^3/3 + \cdots$:

$$\dot{\kappa} \simeq (\lambda - 1)\kappa - \frac{\lambda}{3}\kappa^3, \qquad \kappa_*^2 \simeq \frac{3(\lambda - 1)}{\lambda}. \tag{12}$$

This is a supercritical pitchfork bifurcation. A brief input along $\mathbf{m}$, $\mathbf{h} = \mathbf{m}u(t)$, adds $u(t)$ directly to Equation 11. Starting near zero, a positive or negative pulse can select the corresponding persistent state. Starting in an established state, a switching pulse must be sufficiently strong or long to cross the basin boundary. After the pulse ends, the state retains its sign: the network stores one binary variable.

Connection to Hopfield networks. In this example, $\mathbf{P} = \lambda \mathbf{m}\mathbf{m}^T/N$ is symmetric and resembles a single stored Hopfield pattern. General low-rank networks allow

reading and feedback vectors to differ, extending beyond symmetric associative memory. For the present scalar equation one can define

$$U(\kappa) = \frac{1}{2}\kappa^2 - \lambda \log\cosh\kappa, \qquad \dot{\kappa} = -U'(\kappa), \qquad \frac{dU}{dt} = -\dot{\kappa}^2 \leq 0.$$

This potential describes a collective state evolving in trajectory time; it is distinct from the correlation potential of Lecture 6.

Inputs and outputs: selecting what the network remembers

Let a scalar stimulus $u(t)$ enter along a vector $\mathbf{I}$, and read out the network along $\mathbf{w}$:

$$\dot{\mathbf{x}} = -\mathbf{x} + \mathbf{m}a(t) + \mathbf{I}u(t), \qquad y(t) = \frac{1}{N}\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}(t)). \tag{13}$$

The input drives activity along $\mathbf{I}$, while recurrence drives it along $\mathbf{m}$. Introduce a filtered stimulus v satisfying $\dot{v} = -v+u$. After freely decaying initial components disappear, write

$$\boxed{\mathbf{x} = \mathbf{m}\kappa + \mathbf{I}v, \qquad \dot{\kappa} = -\kappa + \frac{1}{N}\sum_i n_i\phi(m_i\kappa + I_i v).} \tag{14}$$

Substitution verifies this reduction directly. For independent directions $\mathbf{m}$ and $\mathbf{I}$, the current trajectory lies in their plane. The variable v is the filtered input, not generally $u(t)$ itself.

Near the origin, $\phi(x) \simeq x$, giving

$$\dot{\kappa} = -(1-\lambda)\kappa + \beta v, \qquad y \simeq \gamma\kappa + dv, \tag{15}$$

where

$$\lambda = \frac{\mathbf{n}^T\mathbf{m}}{N}, \qquad \beta = \frac{\mathbf{n}^T\mathbf{I}}{N}, \qquad \gamma = \frac{\mathbf{w}^T\mathbf{m}}{N}, \qquad d = \frac{\mathbf{w}^T\mathbf{I}}{N}.$$

These overlaps have distinct meanings. The reading vector $\mathbf{n}$ selects which input enters the recurrent channel through β. The feedback overlap λ controls its persistence, and γ determines how strongly the collective state appears at the output. The term dv is a direct filtered response to the stimulus. An overlap divided by N includes vector magnitudes; it is not just the cosine of an angle.

A minimal selective-integration example

Present two stimulus streams along $\mathbf{I}_A$ and $\mathbf{I}_B$, chosen so

$$\frac{\mathbf{n}^T\mathbf{I}_A}{N} \neq 0, \qquad \frac{\mathbf{n}^T\mathbf{I}_B}{N} = 0.$$

To linear order, stream A enters the recurrent channel while stream B does not. If λ is slightly below one, the selected input leaves a long-lived collective response. Choose a readout with $\mathbf{w}^T\mathbf{m}/N \neq 0$ and negligible direct input overlaps so that the output reports this recurrent signal. For time scales longer than the input filter but shorter than $1/(1-\lambda)$, the collective mode behaves approximately as an integrator of the selected stream.

This provides a simple interpretation of computation: the network selects an input, accumulates or maintains its effect, and exposes the collective state to a readout. The overlap argument is a small-signal description; strong inputs change neuronal gains and can recruit nonlinear interactions.

A brief extension to several collective variables

For R structured pathways and no input or random component, the invariant current subspace is

$$\mathbf{x} = \sum_{a=1}^{R} \mathbf{m}^{(a)} \kappa_a.$$

The collective equations are

$$\boxed{\dot{\kappa}_a = -\kappa_a + \frac{1}{N}\sum_i n_i^{(a)} \phi\left(\sum_b m_i^{(b)} \kappa_b\right).} \tag{16}$$

At the origin of the tanh model, the small overlap matrix

$$B_{ab} = \frac{(\mathbf{n}^{(a)})^T \mathbf{m}^{(b)}}{N}$$

sets the linear dynamics: $\dot{\boldsymbol{\kappa}} = (-\mathbf{I}_R + \mathbf{B})\boldsymbol{\kappa}$. Its off-diagonal entries couple different collective variables. Increasing rank permits richer dynamics, including interacting memory variables and, with appropriate connectivity and nonlinear saturation, oscillations. The number of neurons can be large while the recurrent computation is described by a small system of equations.

What changes when random connectivity is restored?

Return to $\mathbf{J} = g\boldsymbol{\chi} + \mathbf{m}\mathbf{n}^T/N$. The network can be written exactly as

$$\dot{x}_i = -x_i + m_i a_N(t) + \eta_i(t) + h_i(t), \qquad a_N(t) = \frac{1}{N}\sum_j n_j \phi(x_j(t)), \qquad \eta_i(t) = g\sum_j \chi_{ij}\phi(x_j(t)). \tag{17}$$

The structured component supplies coherent feedback along $\mathbf{m}$; the random component supplies heterogeneous recurrent input. Under the large-N assumptions of Lecture 6, DMFT represents this latter input by a self-consistent Gaussian field. The new ingredient is the collective signal $a(t)$, which must also be determined from the activity.

For this lecture, the essential point is that the structured signal is kept explicitly while DMFT describes the random contribution. Random connectivity can introduce activity outside the structured subspace and can generate chaotic fluctuations. Consequently, the exact low-dimensional reduction above applies to the purely low-rank model; with a random component it describes the organized part of a more general activity pattern. We will not derive the extended DMFT equations here.

Population structure and context: a qualitative example

The rank specifies the number of recurrent coordinates, but the distribution of connectivity and input weights across neurons also shapes their dynamics. Dubreuil et al. [2] show how this population structure can support flexible input–output mappings.

The relevant mechanism can be understood from the input sensitivity at a particular operating state. A small change δv in a filtered sensory input changes the rate overlap by

$$\delta a \simeq \left[\frac{1}{N}\sum_i n_i I_i \phi'(x_i)\right] \delta v.$$

The effective sensory coupling therefore depends on neuronal gain $\phi'(x_i)$ as well as the fixed connectivity weights. For tanh, neurons near zero have high gain and strongly saturated neurons have low gain. A contextual input can change these gains by changing the operating state.

For two populations, a useful approximation is

$$\boxed{\beta_{\text{eff}} \simeq f_1 G_1 \beta_1 + f_2 G_2 \beta_2,} \tag{18}$$

where f_p is the population fraction, β_p is its sensory overlap, and G_p is its gain. This approximation treats gain as approximately uniform within each population; more general models retain the weighted average $\mathbb{E}_p[nI\phi'(x)]$ explicitly.

For equal fractions and opposite overlaps, $\beta_1 = b$ and $\beta_2 = -b$, the effective coupling is approximately $b(G_1 - G_2)/2$. Contexts that preferentially reduce the gain of different populations can reverse its sign. With two sensory streams, differential gains can instead change which stream is selected for integration. The synaptic weights remain fixed while their effective influence changes. Context can also change recurrent feedback, so input selection is one aspect of the altered dynamics.

The takeaway is that low dynamical dimension and population structure play complementary roles. Even one recurrent variable can receive different input combinations in different contexts. These populations are defined by their statistical connectivity properties and need not be separate anatomical cell types. A detailed Gaussian closure is unnecessary to understand this mechanism.

Questions and exercises

1. Derive the exact rank-one projection equations. What happens to an initial current component perpendicular to $\mathbf{m}$? Explain why the nonlinear outputs can require several principal components even when one collective coordinate determines the state.

2. For the binary-pattern memory, plot $-\kappa + \lambda \tanh \kappa$ for $\lambda = 0.8, 1.0, 1.5$ and find the stable states. Why is $\lambda = 1$ only a marginal integrator in the linearized model? What does the cubic term predict?

3. Derive Equation 15. Choose two input directions so that only one drives the recurrent channel to linear order. Explain how a readout can report the persistent component rather than the direct input.

4. Explain how differential population gains can reverse an effective sensory coupling without changing the connectivity matrix. Which feature of the neuronal nonlinearity makes this possible?

References

[1] F. Mastrogiuseppe and S. Ostojic, *Linking Connectivity, Dynamics, and Computations in Low-Rank Recurrent Neural Networks*, Neuron **99**, 609–623.e29 (2018). doi:10.1016/j.neuron.2018.07.003.

[2] A. Dubreuil, A. Valente, M. Beiran, F. Mastrogiuseppe, and S. Ostojic, *The role of population structure in computations through neural dynamics*, Nature Neuroscience **25**, 783–794 (2022). doi:10.1038/s41593-022-01088-4.