

Lecture 1:

Firing rate models

Jochen Braun

Otto-von-Guericke-Universität Magdeburg,
Cognitive Biology Group

Theoretical Neuroscience II, SS 2020

Credits:

Dayan & Abbot (2001), "Theoretical Neuroscience", MIT
Press, Chapter 7.

Semester overview

17 lectures will cover four large subjects.

Firing rate models

- ▶ 1. Rate models
- ▶ 2. Feedforward networks
- ▶ 3. Recurrent networks
- ▶ 4. Associative memory
- ▶ 5. State-space analysis

Hebbian plasticity

- ▶ 6. Hebbian plasticity
- ▶ 7. Stable Hebbian plasticity
- ▶ 8. Hebbian models of development

Reinforcement learning

- ▶ 9. Classical conditioning, Rescorla-Wagner rule
- ▶ 10. Temporal difference learning
- ▶ 11. Policy learning
- ▶ 12. Actor-critic models

Representational learning

- ▶ 13. Introduction to causal models
- ▶ 14. K-Means and expectation maximization
- ▶ 15. Principal component analysis
- ▶ 16. Sparse coding
- ▶ 17. Independent component analysis

Course goal

- ▶ **Axiom: function is the key to understanding all aspects of biology, at all levels of scale.**
- ▶ **Theoretical neuroscience seeks possible principles of brain function.**
- ▶ **Course goal is to help you develop a personal, independent judgment about such principles.**
- ▶ **To this end, you will try out and apply these principles in computational exercises.**
- ▶ **Course goal is ambitious, but past experience shows it to be realistic.**
- ▶ **You will not become a theoretician, but competent to interact productively with theoreticians.**



Ms. Zahra Yazdani

Matlab tutorials, online and paperback

- ▶ www.mathworks.com/help/pdf_doc/matlab/getstart.pdf
“MATLAB Primer”
- ▶ de.mathworks.com/support/learn-with-matlab-tutorials.html
“MATLAB Video Tutorials”
- ▶ <https://matlabacademy.mathworks.com/> “Matlab Academy”
- ▶ www.matlab.eng.ed.ac.uk “An interactive introduction to MATLAB”
- ▶ “Matlab: A Practical Introduction to Programming and Problem Solving,” Boston University Paperback.
- ▶ “MATLAB for Neuroscientists,” Associated Press.

Matlab assignments (teams of two):

- ▶ 1. Selective amplification
- ▶ 2. State-space analysis
- ▶ 3. Supervised learning
- ▶ 4. Visual development
- ▶ 5. Rescorla-Wagner rule
- ▶ 6. Policy learning
- ▶ 7. Expectation maximization
- ▶ 8. Principal component analysis

Preconditions for exam registration: (i) all assignments submitted on time, (ii) each received $\geq 50\%$ of points.

Points received count 50% towards final grade!

Lecture 1: Firing rate models

It is not easy to record the activity of many neurons in a population. To analyze and model such activity, we have to simplify. For example, summarize activity in terms of 'firing rate' (spike density of homogeneous population).

*How does population firing change with external input? We analyze this in terms of 'average synaptic current' and 'activation function', and find **two successive low-pass filters**: synaptic time-constant and effective membrane time-constant.*

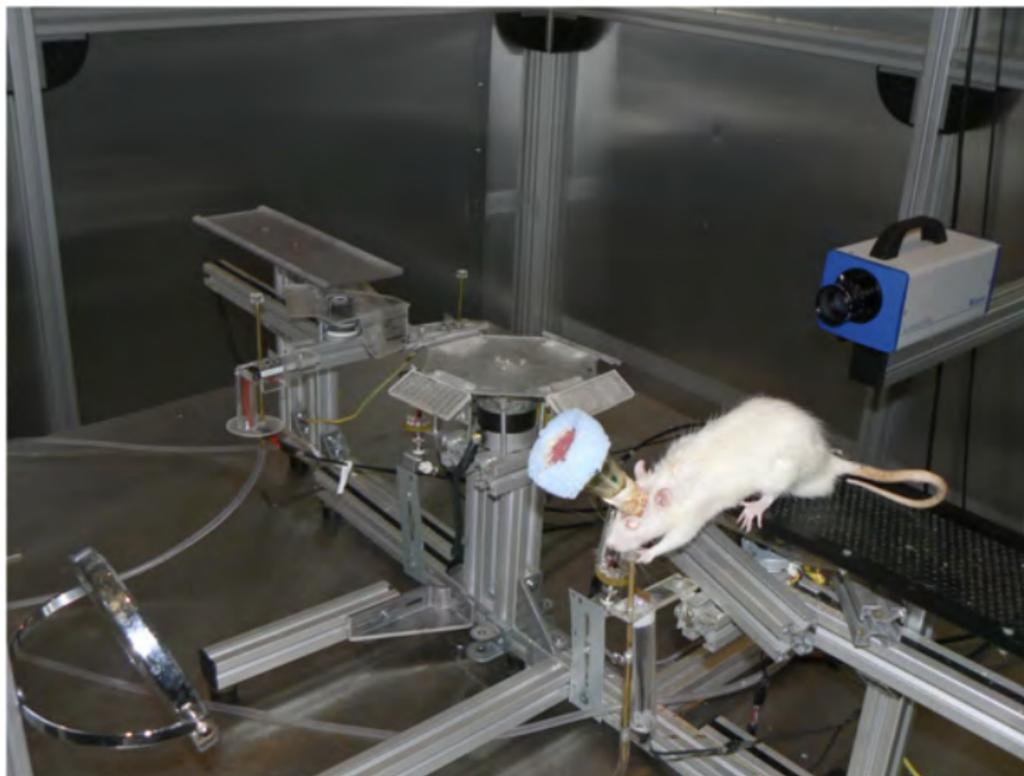
*Lastly, we begin to consider **multiple populations**, linked by synaptic projections. A key role is played by connectivity (synaptic weights), which is conveniently formalized in terms of **Linear Algebra**. Feedforward and recurrent networks behave very differently and require different methods of analysis (see next lectures).*

Parts of lecture

- ▶ 1. Spikes and firing rates
- ▶ 2. Propagating firing rate: one population
- ▶ 3. Propagating firing rate: multiple populations
- ▶ 4. Outlook

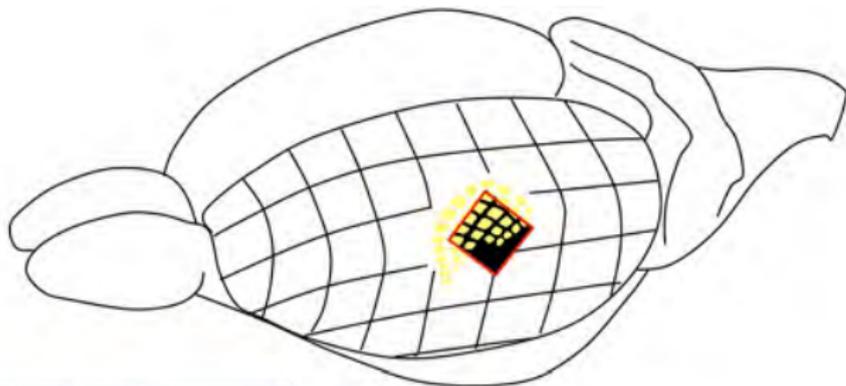
Warning: This is one of the more difficult lectures, because it touches on several different subjects.

1. Spikes and firing rates



Mathew Diamond, SISSA, Trieste

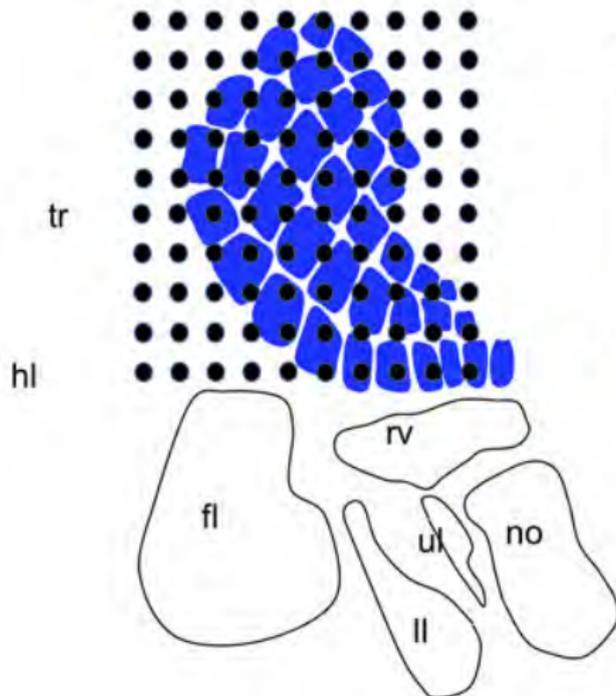
Somatocortical domains of individual whiskers



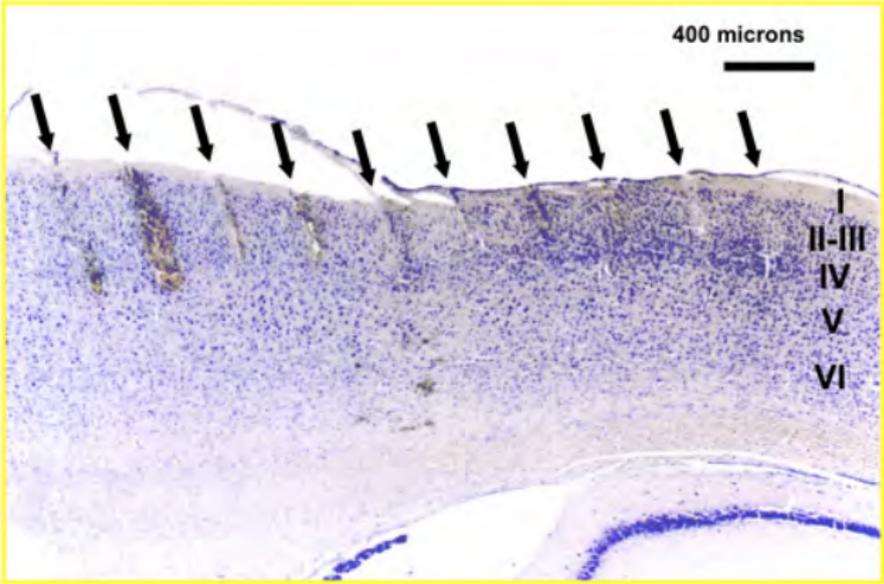
Rat brain, barrel cortex (yellow patches), and electrode array (red square).

Implanted electrode array

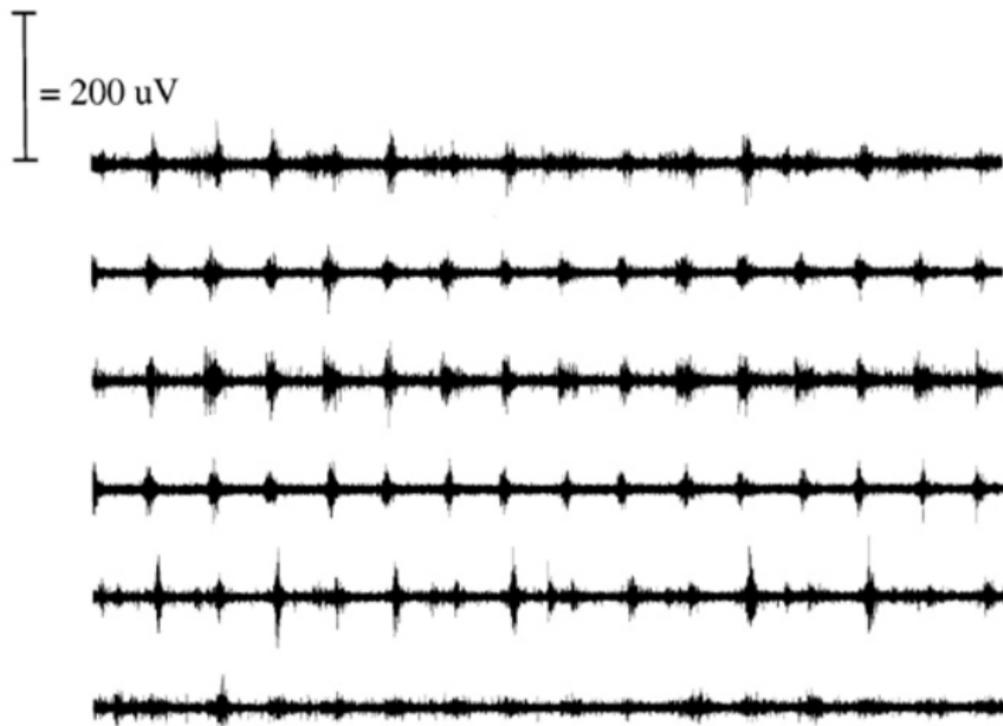
10x10 electrode matrix



Post-mortem reconstruction



MUA from 6 electrodes



Single-unit activity

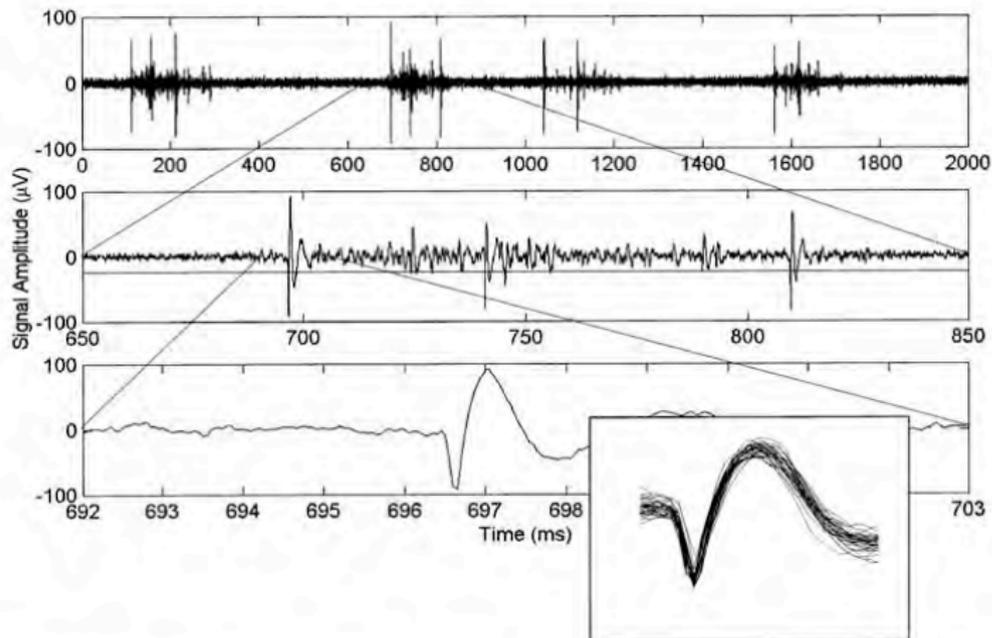
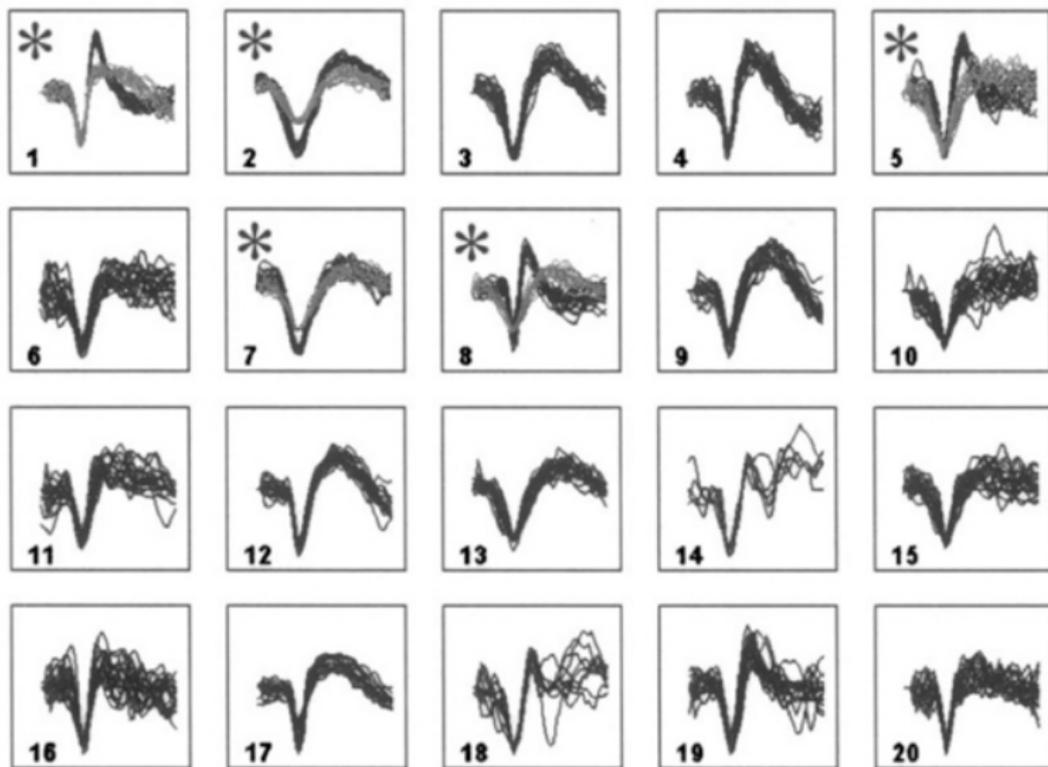


Fig. 3. Discrimination of single-unit activity. A trace of recorded activity is illustrated at increasing temporal magnification. The inset in the bottom plot shows a single unit event waveform isolated by spike-sorting algorithms. Although several units are located within the firing cluster, only the largest unit is separated. S/N of this unit is about 13, slightly higher than the population sample of 11.5 described in Table 1.

Sorting spikes of individual neurons (optional)

E



Whisker-evoked population activity

Within each whisker domain, neuron firing is modulated collectively (trial-by-trial variance is correlated). Surprisingly, spikes remain irregular and asynchronous (see next slide).

P.J. Rousche et al. / Journal of Neuroscience Methods 90 (1999) 57–66

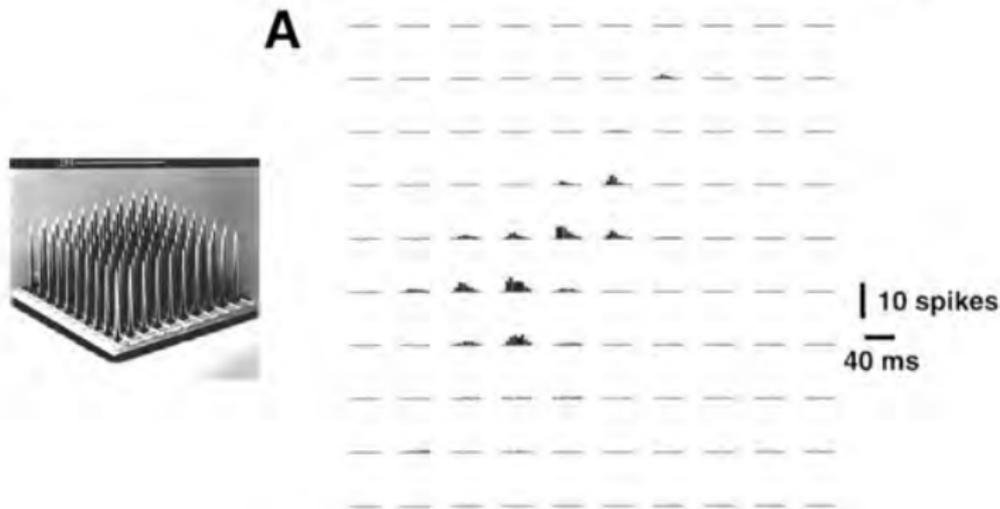
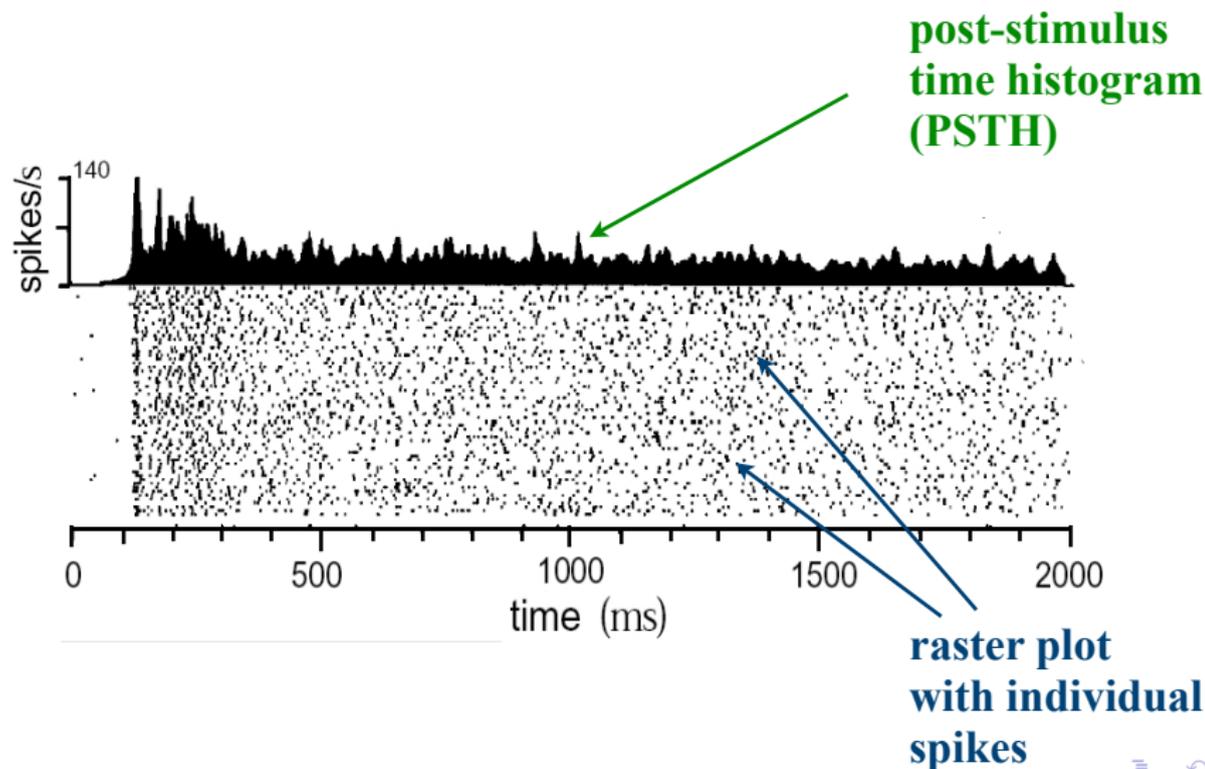


Fig. 6. Reconstructed images of whisker-evoked cortical activity. The data are arranged to represent the 10×10 matrix of electrodes covering a 3.6×3.6 mm field of cortex. (A) PSTH onset corresponds to the beginning of vibrissal deflection (four whiskers—E1–E4). Response values are in events per trial. Twenty-two of the 100 electrodes show some type of response. (B) The data in the PSTHs have been interpolated across

Spike-raster and post-stimulus-time histogram

Spikes are irregular and asynchronous (can occur at any time).

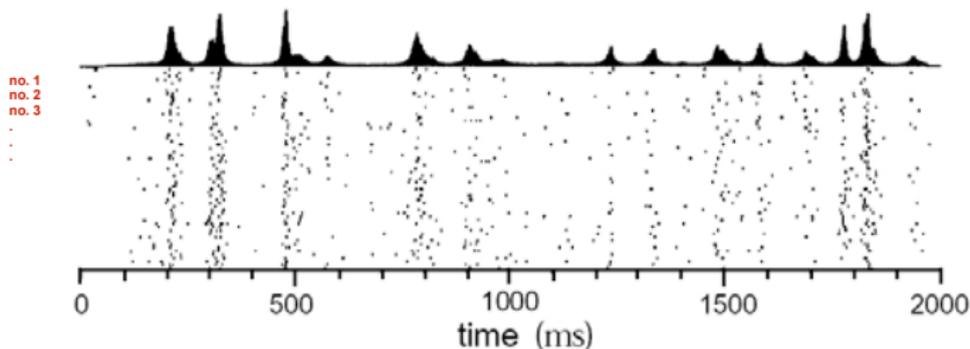


Firing rate defined as 'spike density'

Spike density is average number of spikes per unit time. For example, recording a neuron spiking at 4Hz over a 2s trial, you expect to count 8 spikes. In 50 trials, you expect 400 spikes. Now average 'sliding windows' of different duration over 50 trials:

<i>size</i>	<i>ave. count</i>	<i>comb. size</i>	<i>comb. count</i>	<i>spike density</i>
5 ms	0.02	250 ms	1	4 Hz
20 ms	0.08	1 s	4	4 Hz
50 ms	0.2	2.5 s	10	4 Hz

PSTH with pronounced modulation of firing rate ('spike density'):



Need to simplify

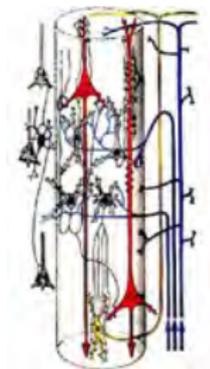
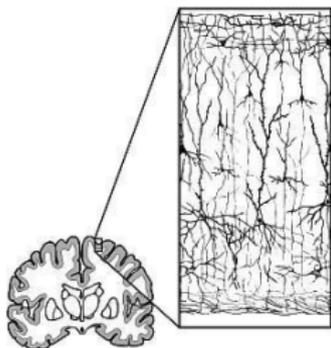
A typical network of cortical neurons involves at a minimum some millions of excitatory and inhibitory neurons (150,000 per mm^3), each with complex dendrites and axons, with many types of membrane channels for Na^+ , K^+ , Ca^{2+} , etc. and receiving input and emitting output through 8,000 synapses on average. Models of this scale are extremely expensive and unwieldy (many time-scales!).

To simplify, we can describe activity of *homogeneous populations* of neurons in terms of *firing rate* (or spike density). Here 'homogeneous' means that firing rates of individual neurons are modulated similarly to the population rate.

Cortical columns: an opportunity to simplify

Within cortical columns, such as whisker domains, neuronal firing is typically homogeneous and modulated similarly by sensory input.

Neurons in cortical columns are densely interconnected and typically fire in an irregular and asynchronous mode.



Ergodicity of homogeneous populations

An average over trials (one neuron, many trials, easy to record!), is often assumed to approximate an average over neurons (many neurons, one trial, far more difficult to record!).

This 'ergodicity' provides an alternative definition for a 'homogeneous population'.

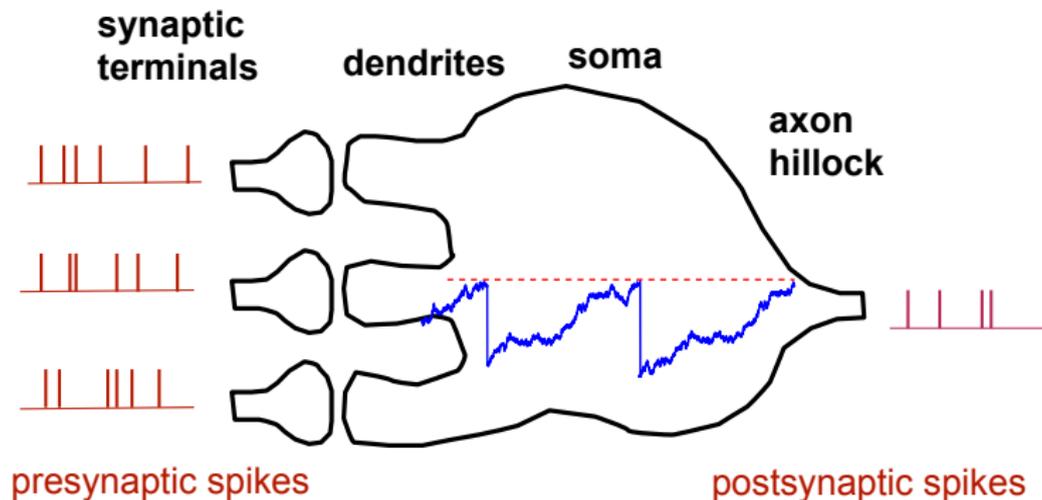
In sensory cortices with columnar organisation, 'ergodicity' is often a reasonable assumption.

Points to note

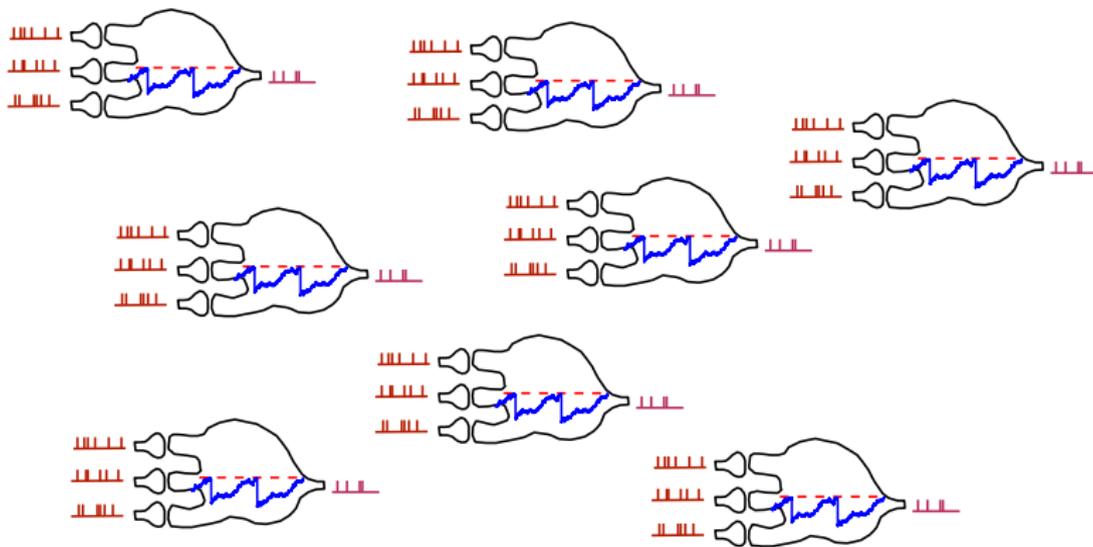
- ▶ Multi-unit recording reveals spikes of many individual neurons ('single units'). Typically, neurons spike irregularly and asynchronously.
- ▶ In columns of sensory cortex, neurons form reasonably homogeneous populations, and firing rates of individual neurons are highly correlated.
- ▶ For such populations, *ergodicity* is often assumed: trial average equals population average.
- ▶ In this case, population activity may be described in terms of 'firing rate' (spike density = average spike count per unit time) and its modulation over time (post-stimulus time histogram, PSTH).

2. Propagating firing rate: one population

Consider a single neuron with pre-synaptic input spikes, total synaptic currents, and post-synaptic output spikes.

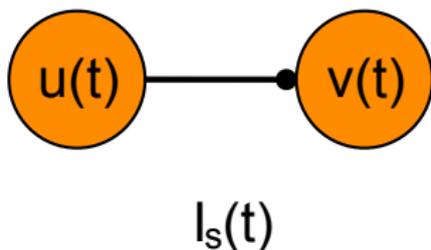


Now consider a homogeneous population of neurons and take an **instantaneous average** over the population, of input spike rates, synaptic currents, and output spike rates.



(The above picture is misleading: individual neurons are not synchronized!)

This gives us mean input rate $u(t)$, mean synaptic current $I_s(t)$, and mean output rate $v(t)$.



Propagation of firing rate: how does input change output?

To understand how $u(t)$ changes $v(t)$, we need to understand how $I_s(t)$ depends on $u(t)$ and how $v(t)$ depends on $I_s(t)$.

Input rate $u(t)$ and synaptic current I_s (advanced)

We assume independent (Poisson distributed) input spikes and individual synaptic currents $i_s(t)$ with an exponential decay

$$i_s(t) = \frac{1}{\tau_s} \exp\left(-\frac{t}{\tau_s}\right), \quad t \geq 0$$

Ignoring scale of u , instantaneous synaptic current is

$$I_s(t) = \int_{-\infty}^t i_s(t-t') u(t') dt' \quad \Leftrightarrow$$

$$\frac{dI_s}{dt} = \frac{u}{\tau_s} - \frac{1}{\tau_s} \int_{-\infty}^t i_s(t-t') u(t') dt' = \frac{1}{\tau_s} (u - I_s)$$

Thus, $I_s(t)$ is a low-pass-filtered version of $u(t)$

$$\tau_s \frac{dI_s(t)}{dt} = -I_s(t) + u(t)$$

Thus, $I_s(t)$ is a low-pass-filtered version of $u(t)$

$$\tau_s \frac{dI_s(t)}{dt} = -I_s(t) + u(t)$$

As long as u is larger (smaller) than I_s , I_s increases (decreases) until it equals u . In other words, $I_s(t)$ follows $u(t)$ with a time-delay (τ_s).

Scaling u properly with synaptic weight w (average electric charge delivered per presynaptic spike = current \times time), we have

$$\tau_s \frac{dl_s}{dt} = -l_s + w u(t)$$

where τ_s is the synaptic time-constant.

We have now linked input rate $u(t)$ and the instantaneous synaptic current $l_s(t)$ of one representative neuron.

Activation function $F(I_s)$

For a constant synaptic current I_s , the output rate will assume a steady-state value v_{ss} , which can be expressed as

$$v_{ss} = F(I_s)$$

where $F()$ is called the *activation function*. The activation function describes the increasing probability of output firing, as synaptic currents move fluctuating membrane potentials in the population closer to threshold.

The activation function $F()$ is monotonically increasing and typically assumes a sigmoidal shape (with threshold, inflection point, and asymptotic saturation).

The activation function $F()$ translates synaptic current into output firing rate.

Output rate $v(t)$ and synaptic current $I_s(t)$

For time-dependent synaptic currents $I_s(t)$, the neuronal membrane capacitance and resistance act as (yet another) low-pass filter. In other words, the *membrane potential* is a low-pass-filtered version of the synaptic current. Accordingly, we expect the time-dependent *output rate* to be a low-pass-filtered version of its (instantaneous) steady-state value v_{ss} :

$$\tau_r \frac{dv(t)}{dt} = -v(t) + v_{ss} = -v(t) + F [I_s(t)]$$

The effective time constant τ_r is not related to membrane constants. Typically, its value is much smaller than the membrane time-constant.

Summary so far

We have now related input rate $v(t)$, synaptic current $I_s(t)$ and output rate $u(t)$ in terms of two dynamic equations (both low-pass filters, or exponential relaxations):

$$\tau_s \frac{dI_s}{dt} = -I_s + w u(t)$$

with synaptic time-constant τ_s and synaptic weight w , and

$$\tau_r \frac{dv(t)}{dt} = -v(t) + F[I_s(t)]$$

with rate time-constant τ_r and activation function $F(s)$.

How accurate are such models?

When output firing is approximately constant, a rate model with a single, fixed value of τ_{eff} provides a good approximation of a more detailed model.

When output firing varies greatly (and sometimes ceases altogether), the value of τ_{eff} must be adjusted (i.e., must change with v) in order to retain a good approximation.

In general, the value of τ_{eff} must be chosen appropriately for the case being studied.

(Adaptive) rate model compared to detailed model

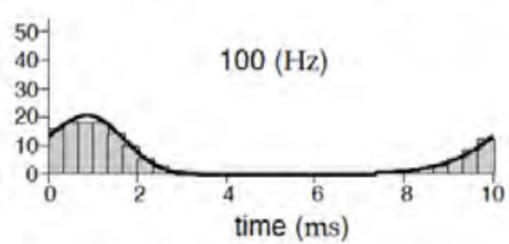
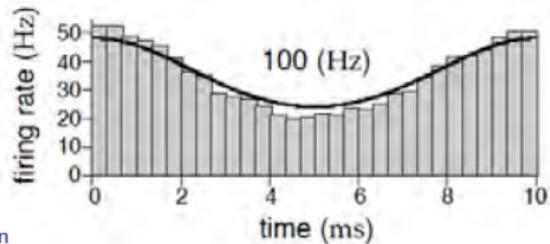
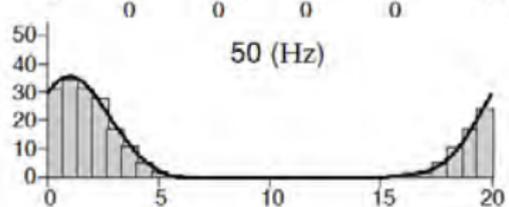
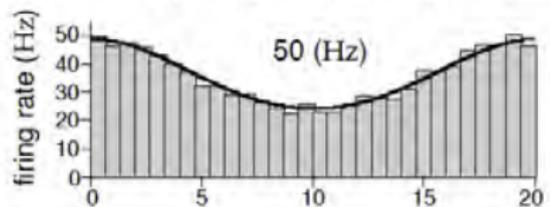
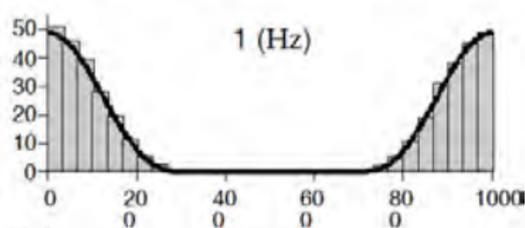
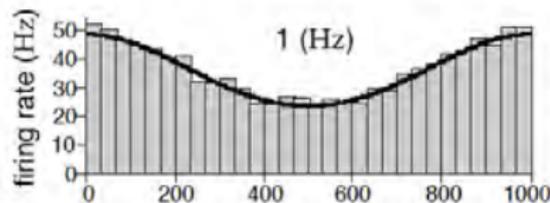
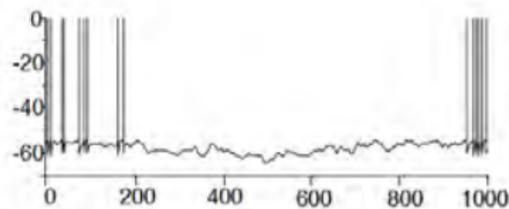
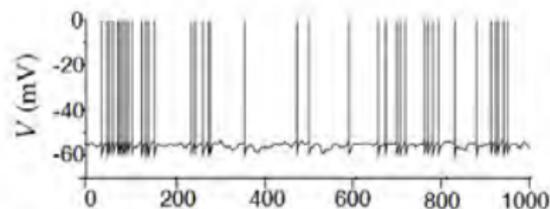


Figure 7.2: Firing rate of an integrate-and-fire neuron receiving balanced excitatory and inhibitory synaptic input and both constant and sinusoidally varying injected current. For the left panels, the constant component of the injected current was adjusted so the firing never stopped during the oscillation of the varying part of the injected current. For the right panel, the constant component was lowered so the firing stopped during part of the cycle. The upper panels show two representative voltage traces of the model cell. The histograms beneath these traces were obtained by binning spikes generated over multiple cycles. They show the firing rate as a function of the time during each cycle of the injected current oscillations. The different rows show 1, 50, and 100 Hz oscillation frequencies for the injected current. The solid curves show the fit of a firing-rate model that involves both instantaneous and low-pass filtered effects of the injected current. (Adapted from Chance *et al*, 2000.)

Points to note

- ▶ We have related collective inputs and outputs of a single neuronal population.
- ▶ The description involves two successive low-pass filters

$$\tau_s \frac{dI_s(t)}{dt} = -I_s(t) + w u(t) \qquad \tau_r \frac{dv(t)}{dt} = -v(t) + F [I_s(t)]$$

- ▶ The first filter represents the synaptic time constant τ_s , and the second filter represents the time-constant τ_r with which firing response follows membrane potential.
- ▶ The description also involves an activation function linking a constant synaptic current to steady-state output firing.

$$v_{ss} = F(I_s)$$

Typically, this is a sigmoidal function, with threshold, inflection point, and saturation.

3. Propagating rates: multiple populations

We have two dynamic equations linking input rate $u(t)$, synaptic current $I_s(t)$, and output rate $v(t)$:

$$\tau_s \frac{dI_s}{dt} = -I_s + w u \qquad \tau_r \frac{dv}{dt} = -v + F(I_s)$$

but seek an even more compact description.

When $\tau_r \gg \tau_s$ (much larger), synaptic current I_s may be replaced by its instantaneous steady-state value $w u$:

$$I_s(t) \approx w u(t) \qquad \Rightarrow \qquad \tau_r \frac{dv}{dt} = -v + F(w u)$$

leaving one *non-linear* dynamic equation. The instantaneous steady-state value of the output rate is

$$v_{ss} = F(w u)$$

Alternatively, when $\tau_r \ll \tau_s$ (much smaller), output rate v may be replaced by its instantaneous equilibrium value $F(I_s)$, leaving

$$\tau_s \frac{dI_s}{dt} = -I_s + w u \quad v \approx F(I_s)$$

Note that the dynamic equation is *linear* in this approximation! This has advantages for some theoretical analyses of network activity.

Again, the instantaneous steady-state value of output firing is

$$v_{ss} = F(w u)$$

Two alternative descriptions of firing population

We typically describe a single neuronal population in terms of its output rate $v(t)$:

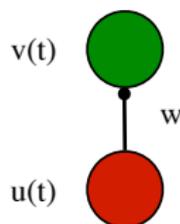
$$\tau_r \frac{dv}{dt} = -v + F[w u(t)]$$

Alternatively, we can choose to describe it in terms of its synaptic current $I_s(t)$:

$$\tau_s \frac{dI_s}{dt} = -I_s + w u \quad v = F(I_s)$$

We are now ready to study the computational capabilities of networks of neuronal populations.

Feedforward one-to-one



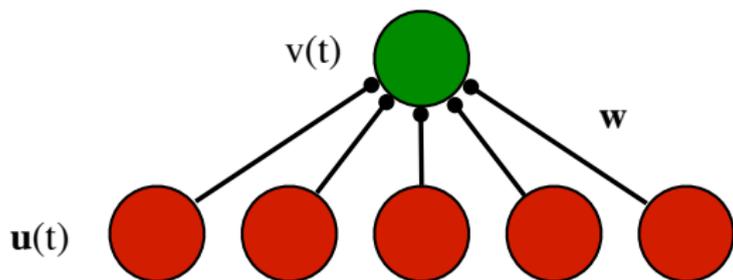
An input unit with rate u drives an output unit with rate v . The relation between input and output rates is approximated by a dynamic equation and the “activation function” $F()$:

$$\tau \frac{dv}{dt} = -v + F(w u)$$

The steady-state firing rate, v_{ss} , is

$$v_{ss} = F(w u)$$

Feedforward many-to-one



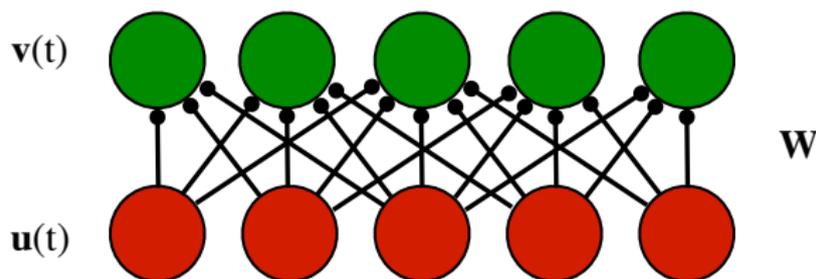
In a feedforward network, N_u input units with rates u_i converge via synapses with weights w_i on a single output units with rate v . For convenience, we collect input units and synaptic weight into vectors \mathbf{u} and \mathbf{w} . The output rate is then approximated by

$$\tau \frac{dv}{dt} = -v + F(\mathbf{w} \cdot \mathbf{u}) \quad \text{or} \quad \tau_r \frac{dv}{dt} = -v + F\left(\sum_{i=1}^{N_u} w_i u_i\right)$$

The steady-state firing rate, v_{∞} , is

$$v_{ss} = F(\mathbf{w} \cdot \mathbf{u})$$

Feedforward many-to-many



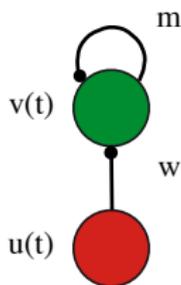
In a feedforward network, N_u input units with rates u_i converge via synapses with weights w_{io} onto N_v output units with rates v_o . For convenience, input and output units are collected in vectors \mathbf{u} and \mathbf{v} , and the synaptic weights into a matrix \mathbf{W} . The output rate is then approximated by

$$\tau \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{F}(\mathbf{W} \cdot \mathbf{u}) \quad \text{or} \quad \tau_r \frac{dv_o}{dt} = -v_o + F\left(\sum_{i=1}^{N_u} w_{io} u_i\right)$$

The steady-state firing rate, \mathbf{v}_{ss} , is

$$\mathbf{v}_{ss} = \mathbf{F}(\mathbf{W} \cdot \mathbf{u})$$

Recurrent one-to-one

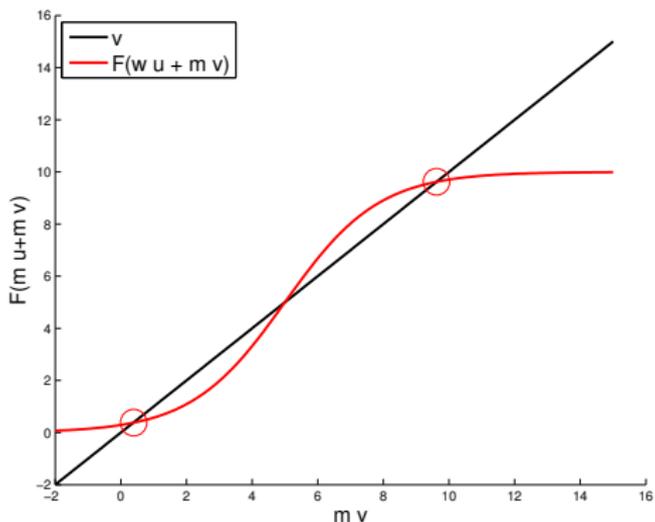


An input unit with rate $u(t)$ drives an output unit with rate $v(t)$, which recurrently feeds back to itself. The relation between input and output rates is approximated by

$$\tau \frac{dv}{dt} = -v + F(w u + m v)$$

To compute the steady-state firing rate, v_{SS} , we need to solve

$$v_{SS} = F(w u + m v_{SS})$$

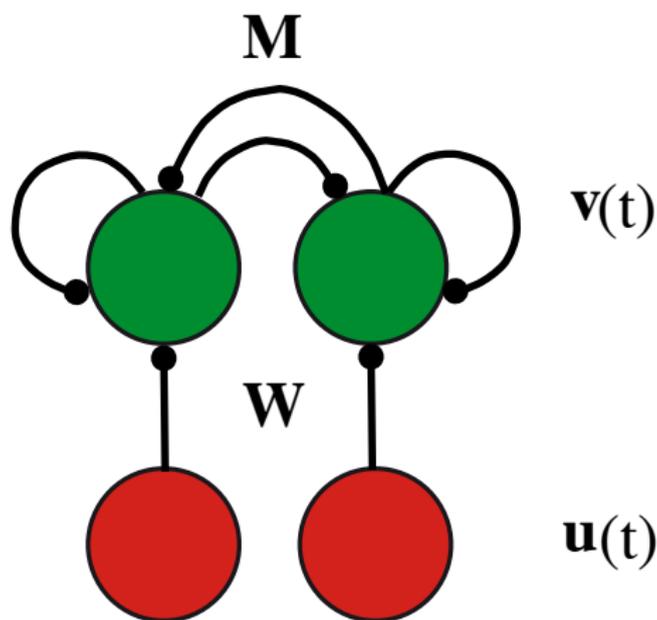


Graphical solution of

$$v_{SS} = F(w u + m v_{SS})$$

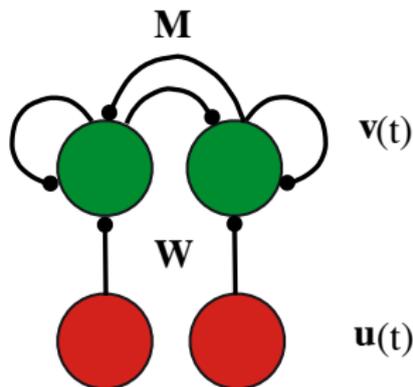
with two steady-state rates v_{SS} (red circles). (The intermediate intersection is unstable). We return to this topic in Lecture 5.

Recurrent many-to-many



$$\tau \frac{dv_1}{dt} = -v_1 + F [w_1 u_1 + m_{11} v_1 + m_{12} v_2]$$

$$\tau \frac{dv_2}{dt} = -v_2 + F [w_2 u_2 + m_{21} v_1 + m_{22} v_2]$$



$$\tau \frac{dv_1}{dt} = -v_1 + F [w_1 u_1 + m_{11} v_1 + m_{12} v_2]$$

$$\tau \frac{dv_2}{dt} = -v_2 + F [w_2 u_2 + m_{21} v_1 + m_{22} v_2]$$

$$\tau \begin{bmatrix} \frac{dv_1}{dt} \\ \frac{dv_2}{dt} \end{bmatrix} = - \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + F \left\{ \begin{bmatrix} w_1 & 0 \\ 0 & w_2 \end{bmatrix} \cdot \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \right\}$$

$$\tau \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{F} [\mathbf{W} \cdot \mathbf{u} + \mathbf{M} \cdot \mathbf{v}]$$

Steady-state firing rate vector

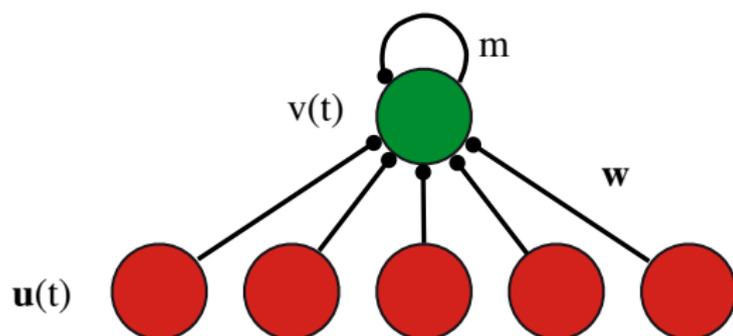
Any steady-state firing rate vector would have to satisfy

$$\mathbf{v}_{ss} = \mathbf{F}(\mathbf{W} \cdot \mathbf{u} + \mathbf{M} \cdot \mathbf{v}_{ss})$$

A steady-state implies that, for a certain activity vector \mathbf{v}_{ss} , the excitatory and inhibitory projections to every population o exactly balance the relaxation terms $-v_o$, so that the time-derivative \dot{v}_o is zero and activity v_o remains stable.

Determining such vectors (and whether they exist at all) requires more advanced methods, involving eigenvectors and eigenvalues of recurrent connectivity \mathbf{M} . We shall return to this issue in Lectures 4 and 5.

Recurrent many-to-one (optional)



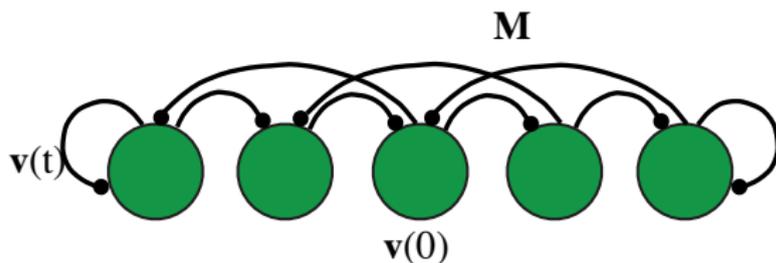
N_u input units with rates u_i project with weights w_i to a single output unit with rate v . The output unit also projects to itself with weight m :

$$\tau \frac{dv}{dt} = -v + F(\mathbf{w} \cdot \mathbf{u} + m v) \Leftrightarrow \tau_r \frac{dv}{dt} = -v + F\left(m v + \sum_{i=1}^{N_u} w_i u_i\right)$$

To compute the steady-state firing rate, v_{ss} , we need to solve

$$v_{ss} = F(\mathbf{w} \cdot \mathbf{u} + m v_{ss})$$

Recurrent many-to-many (optional)



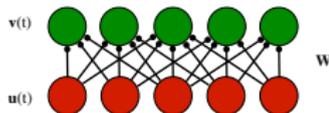
N_u input units with rates u_i project with weights w_{io} to N_v output units with rates v_o . The output units also project with weights $m_{o'o}$ to each other:

$$\tau \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{F}(\mathbf{W} \cdot \mathbf{u} + \mathbf{M} \cdot \mathbf{v}) \quad \Leftrightarrow$$

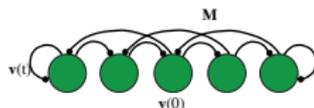
$$\Leftrightarrow \quad \tau_r \frac{dv_o}{dt} = -v_o + F \left[\sum_{i=1}^{N_u} w_{io} u_i + \sum_{o'=1}^{N_v} m_{o'o} v_{o'} \right]$$

Points to note

- ▶ In describing networks with multiple populations of neurons, *connectivity* assumes a key role. Typically, we collect all connection weights into a *connectivity matrix*.
- ▶ In feedforward networks, all projections are in the same 'feedforward' direction:



- ▶ Recurrent networks involve both 'feedforward' and 'feedback' connections. Here we consider only *one-layer recurrent* networks, with 'lateral' connections in the output layer:



- ▶ Feedforward and recurrent networks behave quite differently and require different methods of analysis.

4. Outlook

- ▶ **Lecture 2:** Feedforward networks for classification and for coordinate transforms.
- ▶ **Lecture 3 & Exercise 1:** Recurrent networks for selective amplification.
- ▶ **Lecture 4:** Recurrent networks for associative memory.
- ▶ **Lecture 5 & Exercise 2:** State-space analysis of small recurrent networks.
- ▶ **Exercise 3:** Supervised learning in feedforward network.