CSE/NEUBEH 528

Lecture 9: Computation by Networks

(Chapter 7)

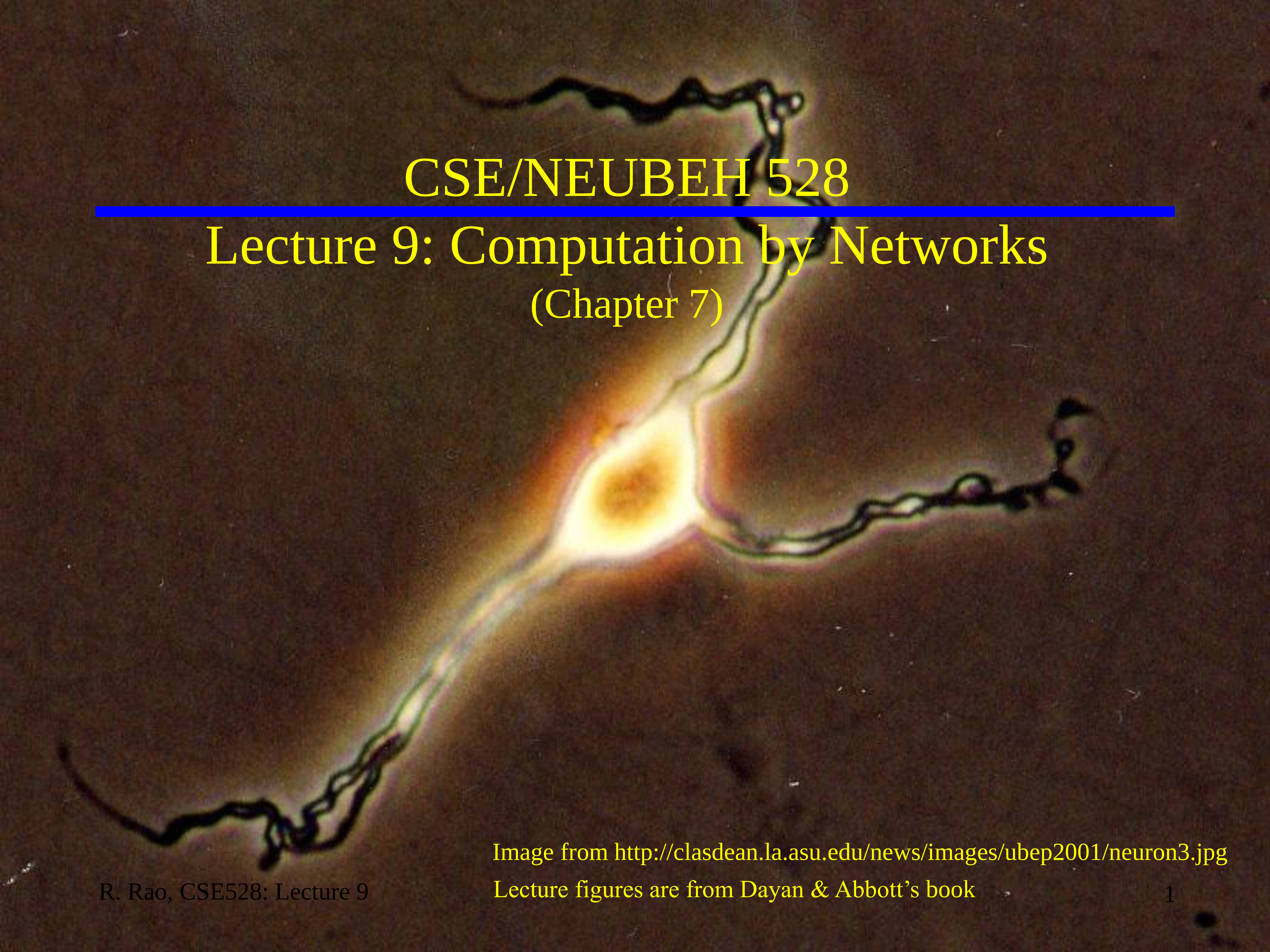


Image from http://clasdean.la.asu.edu/news/images/ubep2001/neuron3.jpg

Lecture figures are from Dayan & Abbott's book

Course Summary (thus far)

- Neural Encoding
  - What makes a neuron fire? (STA, covariance analysis)
  - Poisson model of spiking
- Neural Decoding
  - Spike-train based decoding of stimulus
  - Stimulus Discrimination based on firing rate
  - Population decoding (Bayesian estimation)
- Single Neuron Models
  - RC circuit model of membrane
  - Integrate-and-fire model
  - Conductance-based Models

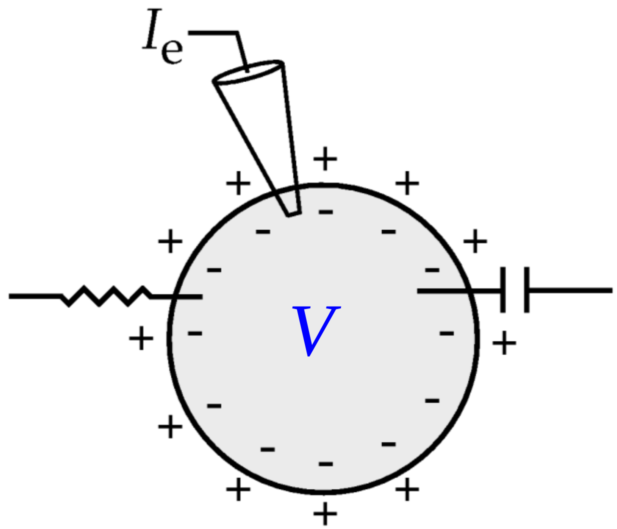
Today's Agenda

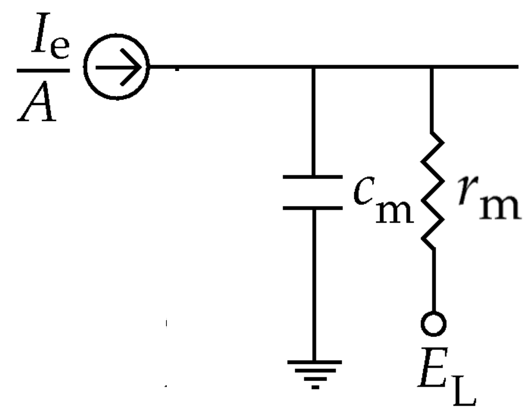
- Computation in Networks of Neurons
  - From spiking to firing-rate based networks
  - Feedforward Networks
  - Linear Recurrent Networks

Modeling Networks of Neurons

- Option 1: Use *spiking* neurons
  - *Advantages*: Model computation and learning based on:
    - Spike Timing
    - Spike Correlations/Synchrony between neurons
  - *Disadvantages*: Computationally expensive
- Option 2: Use neurons with *firing-rate outputs (real valued outputs)*
  - *Advantages*: Greater efficiency, scales well to large networks
  - *Disadvantages*: Ignores spike timing issues
- Question: How are these two approaches related?

Flashback Membrane Model





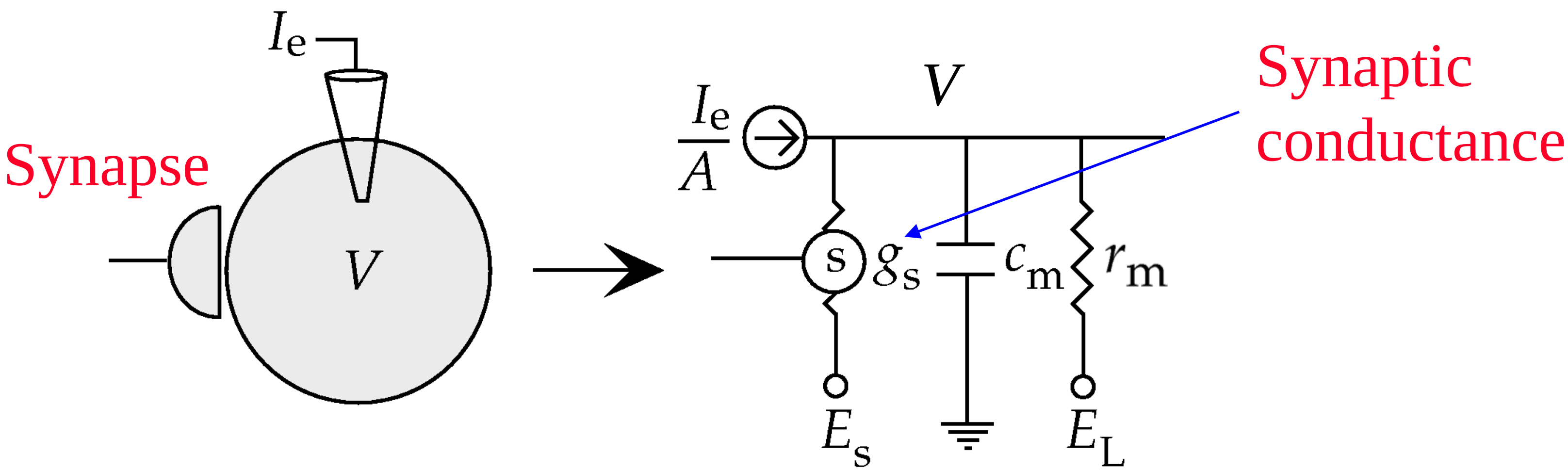
$$c_m \frac{dV}{dt} = -\frac{(V - E_L)}{r_m} + \frac{I_e}{A}, \text{ or equivalently}$$

$\tau_m = r_m c_m = R_m C_m$
membrane time constant

$$\tau_m \frac{dV}{dt} = -(V - E_L) + I_e R_m$$

Flashback

Modeling Synaptic Inputs from other Neurons



$$\tau_m \frac{dV}{dt} = -(V - E_L) - r_m g_s (V - E_s) + I_e R_m$$

$$g_s = g_{s,\max} P_{rel} P_s$$

P_s ← Probability of postsynaptic channel opening (= fraction of channels opened)

P_{rel} ← Probability of transmitter release given an input spike

Basic Synapse Model

- Assume $P_{rel} = 1$
- Model the effect of a single spike input on P_s
- Kinetic Model: $$\text{Closed} \underset{\beta_s}{\overset{\alpha_s}{\rightleftharpoons}} \text{Open}$$

$$\frac{dP_s}{dt} = \alpha_s(1 - P_s) - \beta_s P_s$$

Opening rate

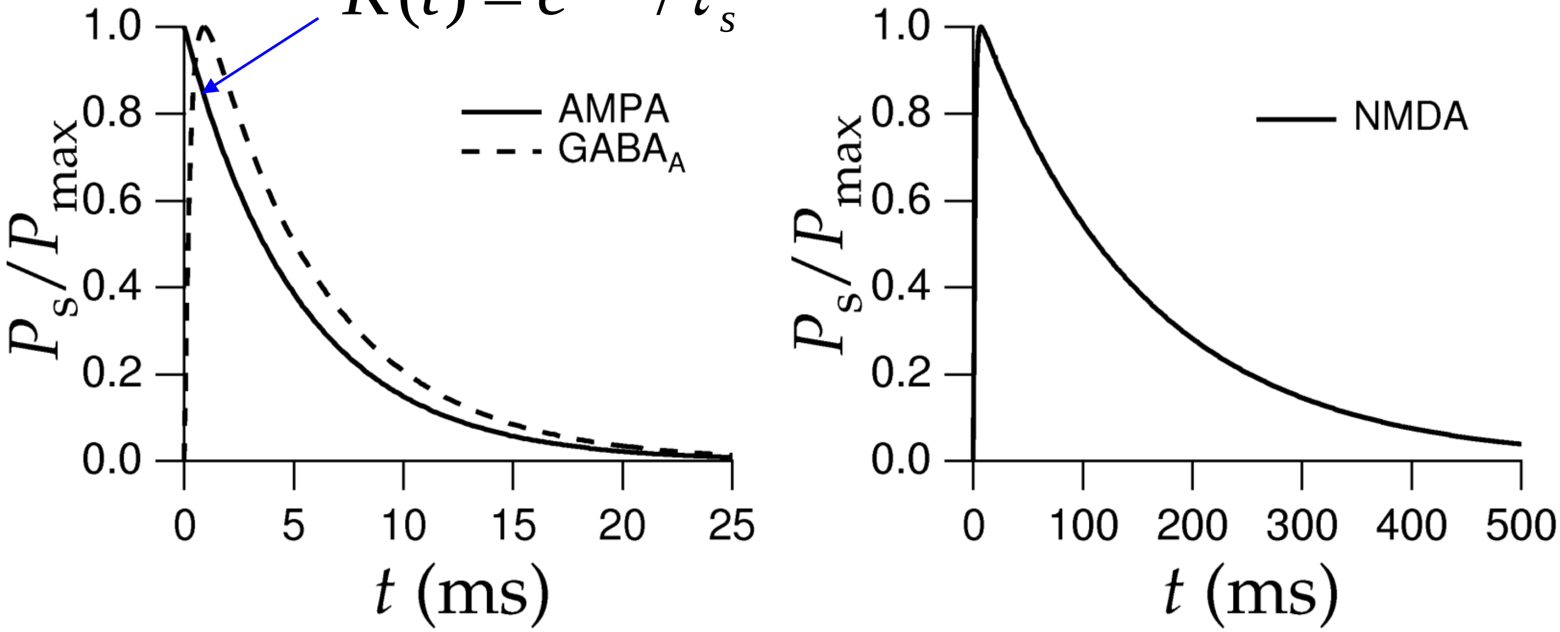
Fraction of channels closed

Closing rate

Fraction of channels open

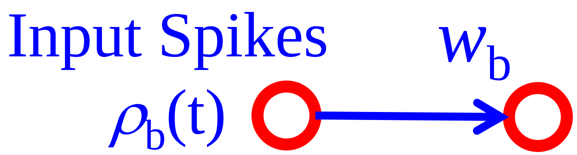
Synaptic Filter and Postsynaptic Data

$$K(t) = e^{-\frac{t}{\tau_s}} / \tau_s$$



Exponential function $K(t)$ gives reasonable fit to biological data (other options: difference of exponentials, "alpha" function)

Modeling a Synaptic Input to a Neuron



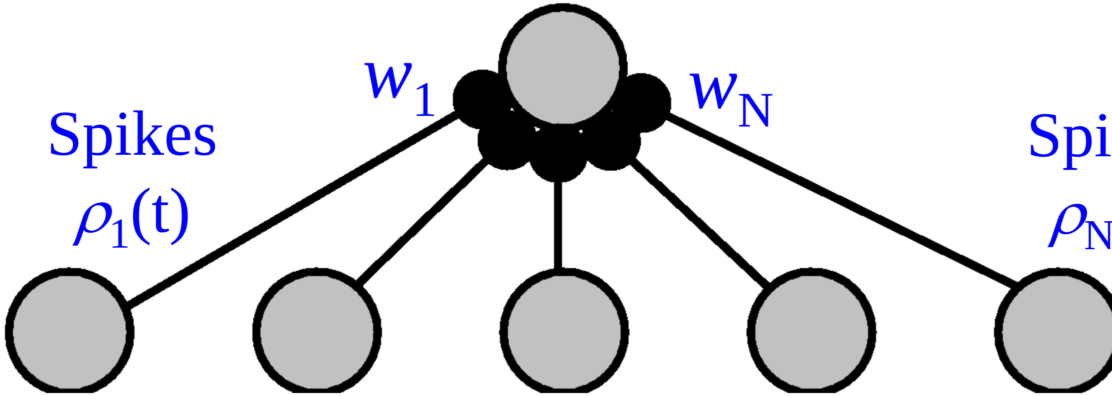
Filter for synapse *b*: $K(t) = \frac{1}{\tau_s} e^{-\frac{t}{\tau_s}}$

Synaptic current: $$I_b(t) = w_b \int_{-\infty}^{t} K(t-\tau)\rho_b(\tau)d\tau$$

where w_s is a synaptic weight and $\rho_b(\tau)$ is the input spike train:

$\rho_b(\tau) = \Sigma_i \, \delta(\tau - t_i)$ (t_i are the input spike times)

From Spiking to Firing Rate Models



Current at synapse *b*

$$I_b(t) = w_b \int_{-\infty}^{t} K(t-\tau)\rho_b(\tau)d\tau$$

Spike train $\rho_b(t)$

$$\approx w_b \int_{-\infty}^{t} K(t-\tau)u_b(\tau)d\tau$$

Firing rate $u_b(t)$

Total synaptic current

$$I_s(t) = \sum_b I_b(t)$$

Synaptic Current Dynamics

- If synaptic kernel K is exponential: $K(t) = \frac{1}{\tau_s} e^{-\frac{t}{\tau_s}}$

  Differentiating $I_s(t) = \sum_b I_b(t) = \sum_b w_b \int_{-\infty}^{t} K(t-\tau) u_b(\tau) d\tau$

  We get $$\tau_s \frac{dI_s}{dt} = -I_s + \sum_b w_b u_b$$
  $$= -I_s + \mathbf{w} \cdot \mathbf{u}$$

Output Firing-Rate Dynamics

- How is the output firing rate v related to synaptic inputs?

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

- Looks very much like membrane dynamics equation:

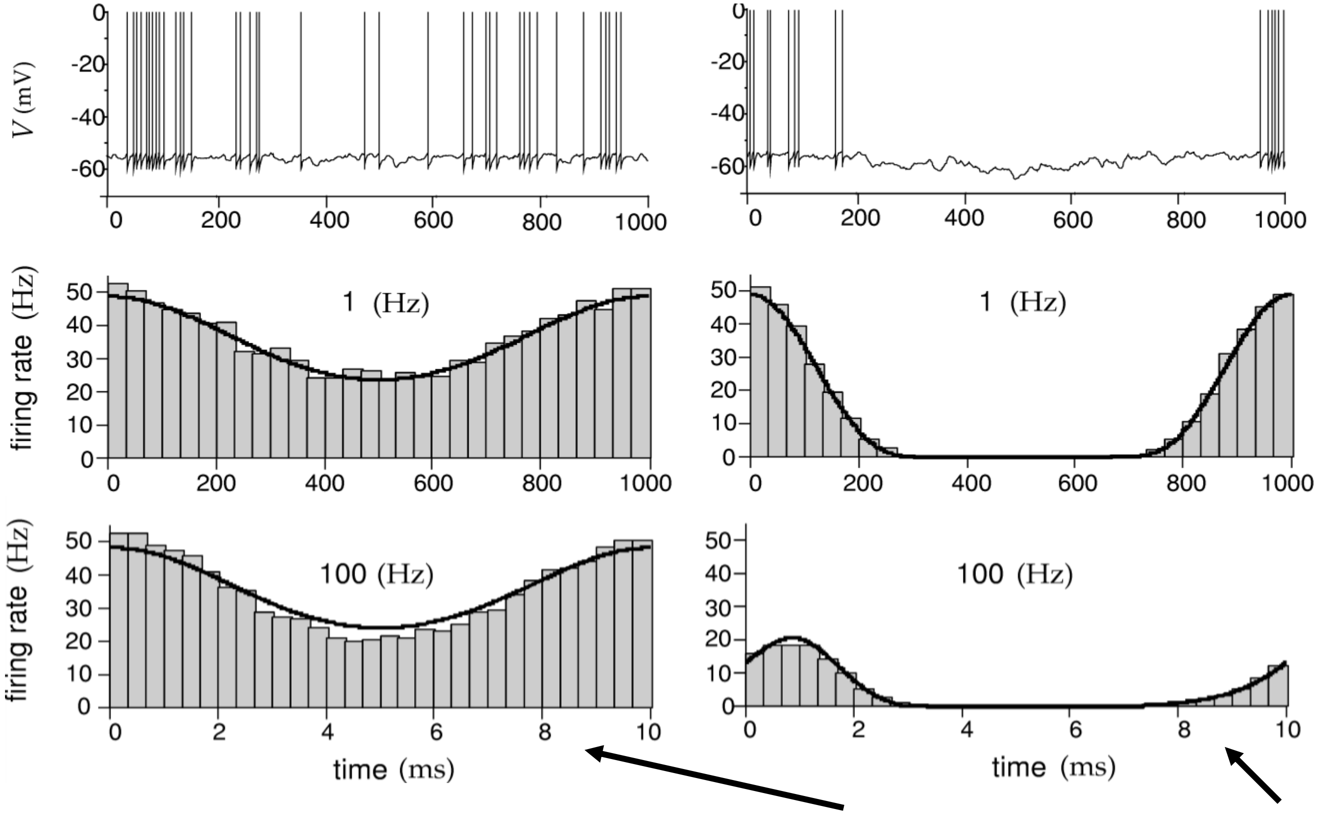
$$\tau_m \frac{dV}{dt} = -(V - E_L) + I_e R_m$$

- On-board derivations of special cases obtained from comparing τ_r and τ_s …

  (see also pages 234-236 in the text)

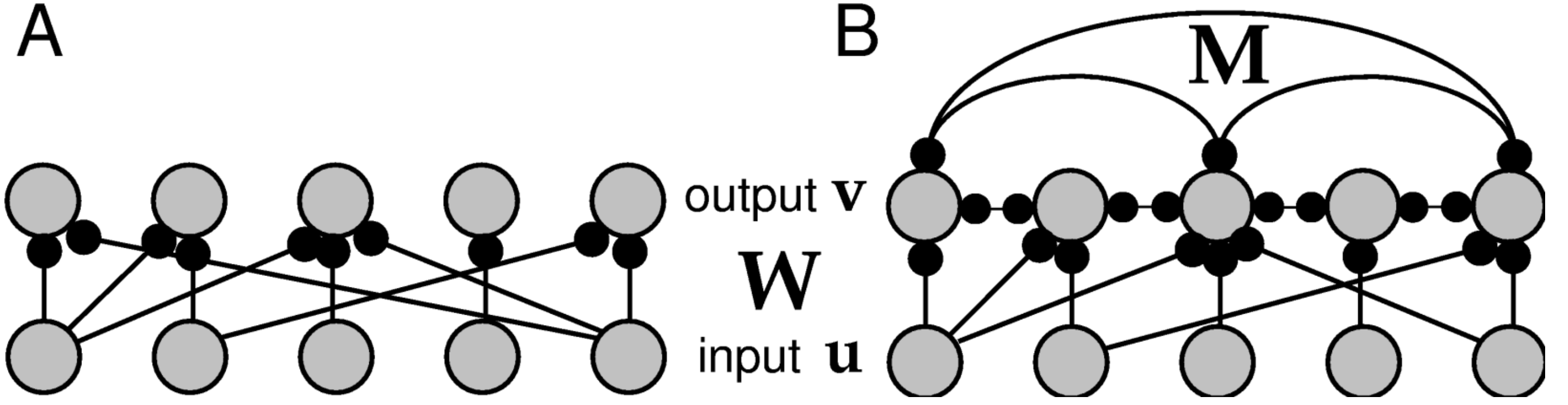
How good are the Firing Rate Models?

Input $I(t) = I_0 + I_1\cos(\omega t)$



Firing rate model v(t) = F(I(t)) describes this well but not this case

Feedforward versus Recurrent Networks

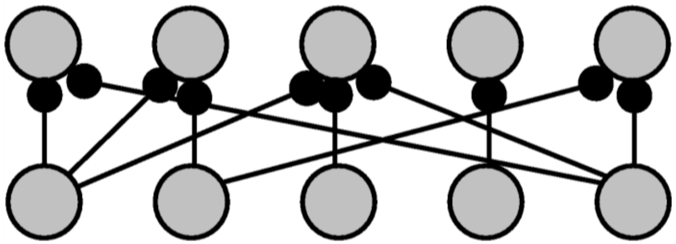


$$\tau \frac{d\mathbf{v}}{dt} = -\mathbf{v} + F(\mathrm{W}\mathbf{u} + \mathrm{M}\mathbf{v})$$

Output Decay Input Feedback

For feedforward networks, matrix M = 0

Example: Linear Feedforward Network



Dynamics: $\tau \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathrm{W}\mathbf{u}$

Steady State (set $d\mathbf{v}/dt$ to 0): $\mathbf{v}_{ss} = \mathrm{W}\mathbf{u}$

$$\mathrm{W} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & 0 & -1 \end{bmatrix}$$

$$\mathbf{u} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 1 \end{bmatrix}$$

What is $\mathbf{v}_{ss}$?

Linear Feedforward Network

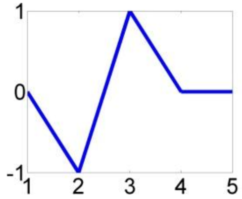
$$\mathbf{v}_{ss} = \mathrm{W}\mathbf{u} = \begin{bmatrix} 1 & 0 & 0 & 0 & -1 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \\ 1 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

What is the network doing?

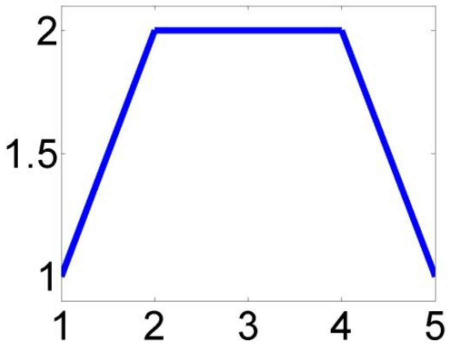
Linear Filtering for Edge Detection

$$\text{Filter} = \begin{bmatrix} 0 & -1 & 1 & 0 & 0 \end{bmatrix}$$

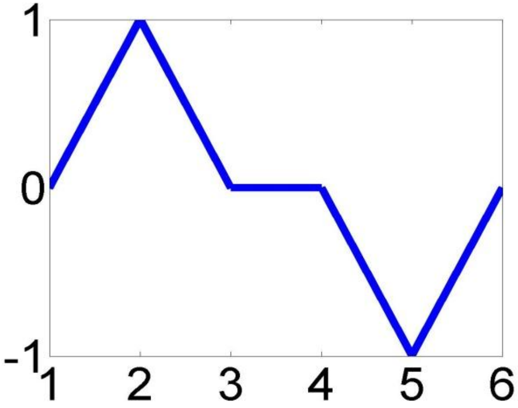
(and shifted versions)



$$\text{Input} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 2 \\ 1 \end{bmatrix} \quad \text{Output} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}$$

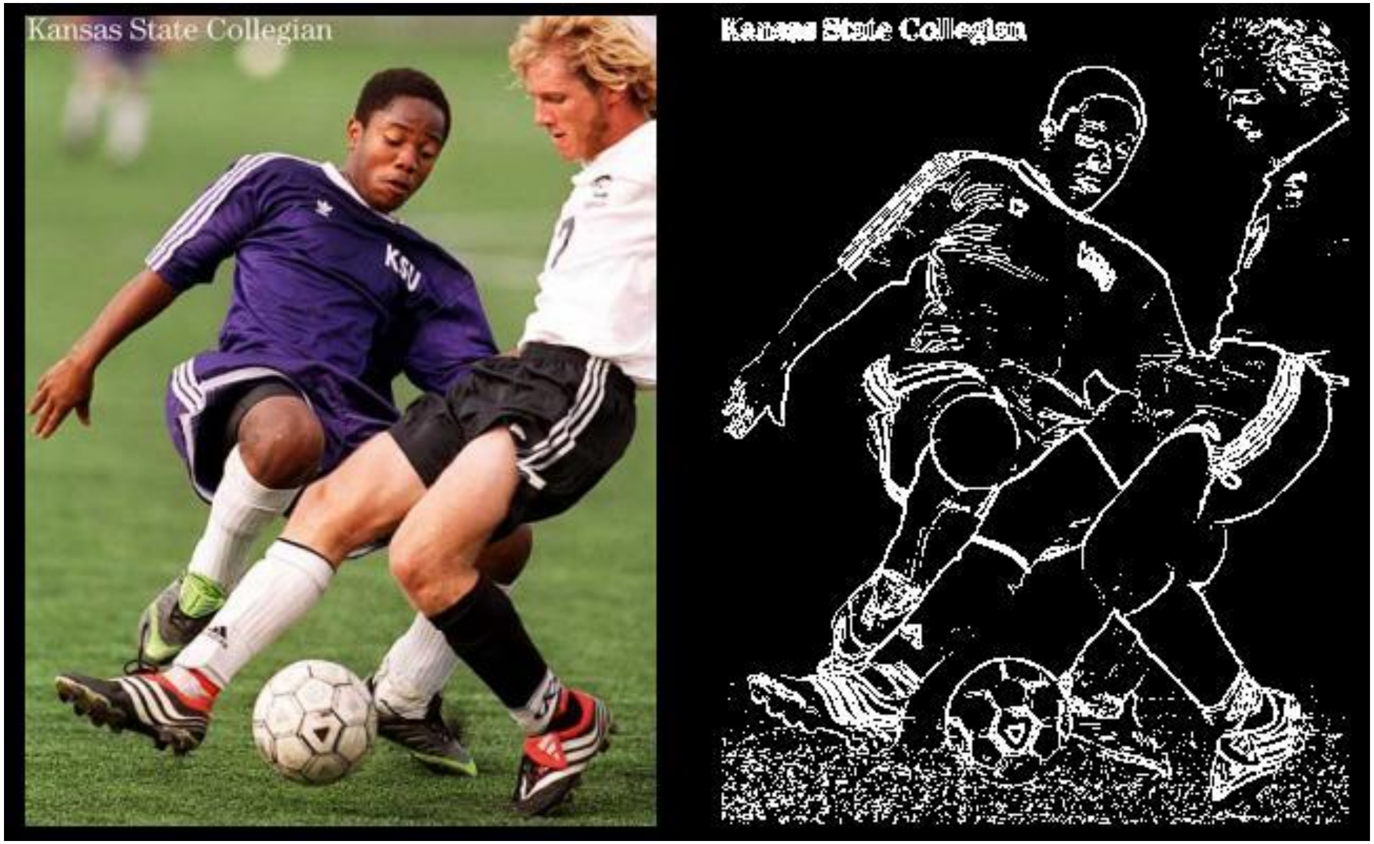


Input



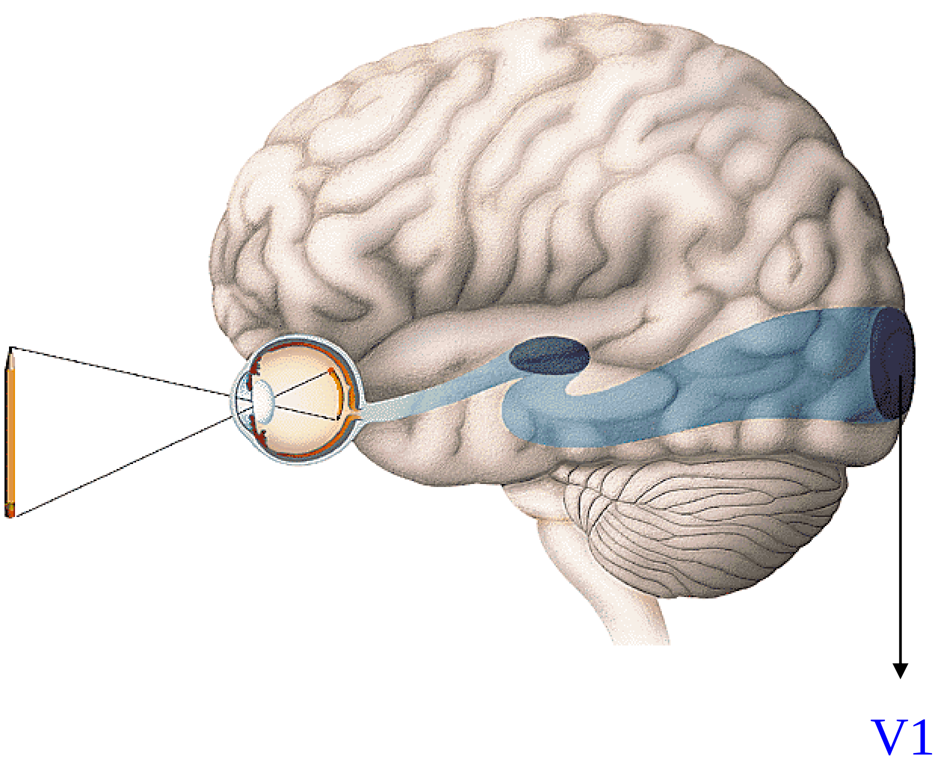
Output

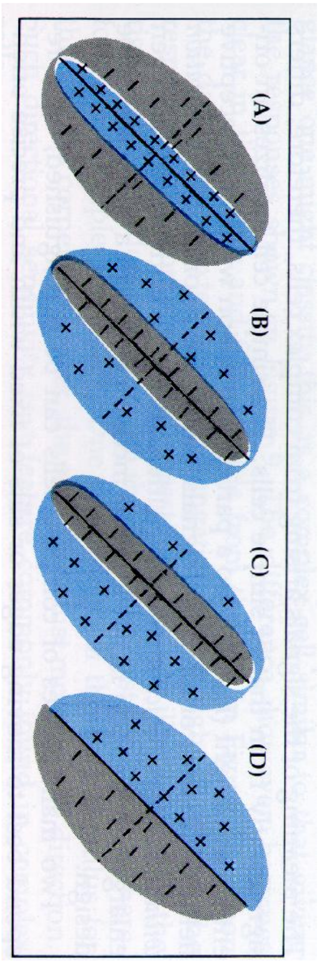
Example of Edge Detection in a 2D Image



http://www.alexandria.nu/ai/blog/entry.asp?E=51

Edge detectors in the visual system

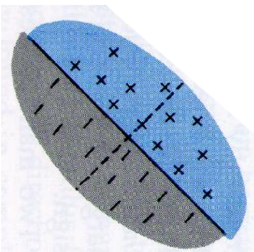




(From Nicholls et al., 1992)

Examples of receptive fields in primary visual cortex (V1)

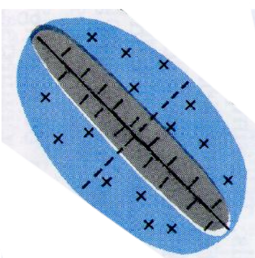
Filtering network is computing derivatives!



$$\begin{bmatrix} 0 & -1 & 1 & 0 & 0 \end{bmatrix}$$

$$\frac{df}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

$$\text{Discrete approximation} \approx f(x+1) - f(x)$$



$$\begin{bmatrix} 0 & 1 & -2 & 1 & 0 \end{bmatrix}$$

$$\frac{d^2 f}{dx^2} = \lim_{h \to 0} \frac{f'(x+h) - f'(x)}{h}$$

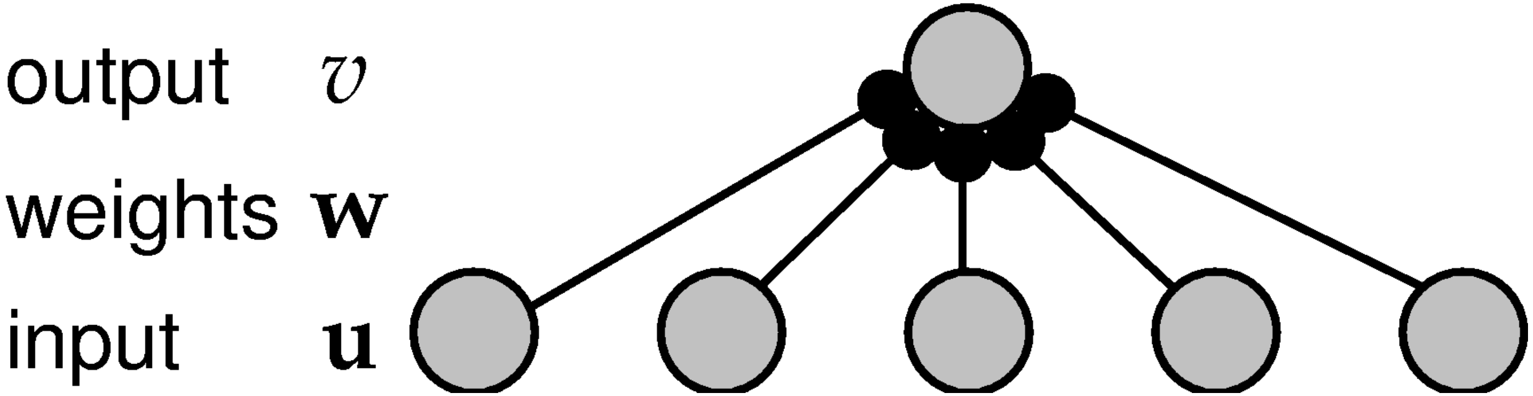
$$\text{Disc. approx.} \approx \big(f(x+1) - f(x)\big) - \big(f(x) - f(x-1)\big)$$

$$= f(x+1) - 2f(x) + f(x-1)$$

Feedforward Networks: Example 2

Coordinate Transformation

Output: Premotor Cortex Neuron with Body-Based Tuning Curves

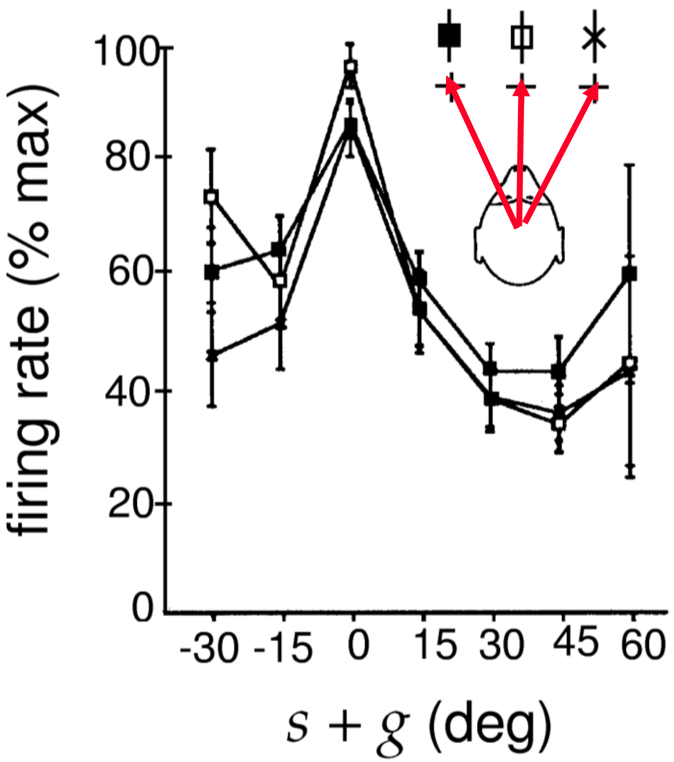


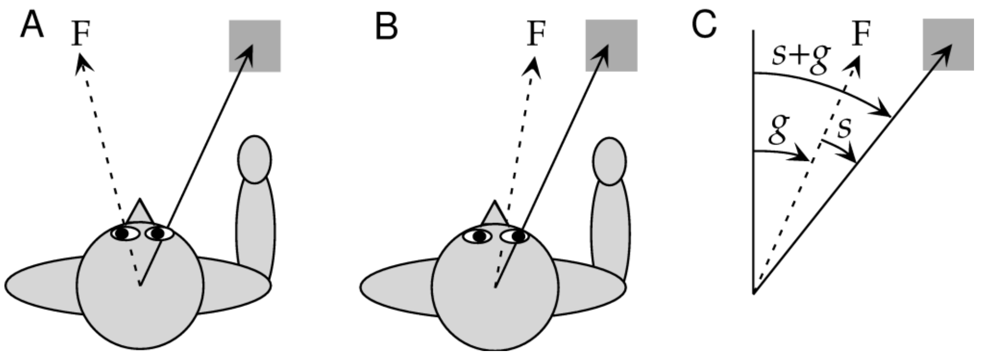
Input: Area 7a Neurons with Gaze-Dependent Tuning Curves

(From Section 7.3 in Dayan & Abbott)

Output of Coordinate Transformation Network

Head fixed;
gaze shifted to g_1 g_2 g_3



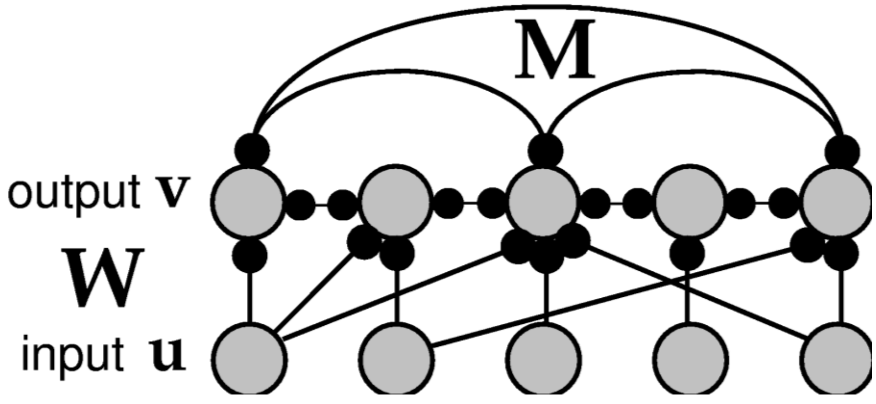


Same tuning curve regardless of gaze angle

Premotor cortex neuron responds to stimulus location *relative to body*, not retinal image location

(See section 7.3 in Dayan & Abbott for details)

Linear Recurrent Networks



$$\tau \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathrm{W}\mathbf{u} + \mathrm{M}\mathbf{v}$$

Output Decay Input Feedback

Next Class: Recurrent Networks

- To Do:
  - Homework 2
  - Choose final project topic and partner(s)