Modeling Networks of Neural Connections

Nathan Crock

Sources from Dayan and Abbott text and Rajesh P.N. Rao's course notes and slides

Summary

- We will begin by constructing a model of chemical synaptic activity between neurons
- We will move from a spiking model to a firing rate model
- Next is exploration of Linear Neural Networks (LNN)
- Then LNNs with symmetric recurrency
- Then we look at Nonlinear NNs with symmetric recurrency
- Lastly we will explore the dynamics of Nonlinear NNs with nonsymmetric recurrency

Synapses

- Point of connection between neurons

- Modeling synapses is key to modeling networks of neurons

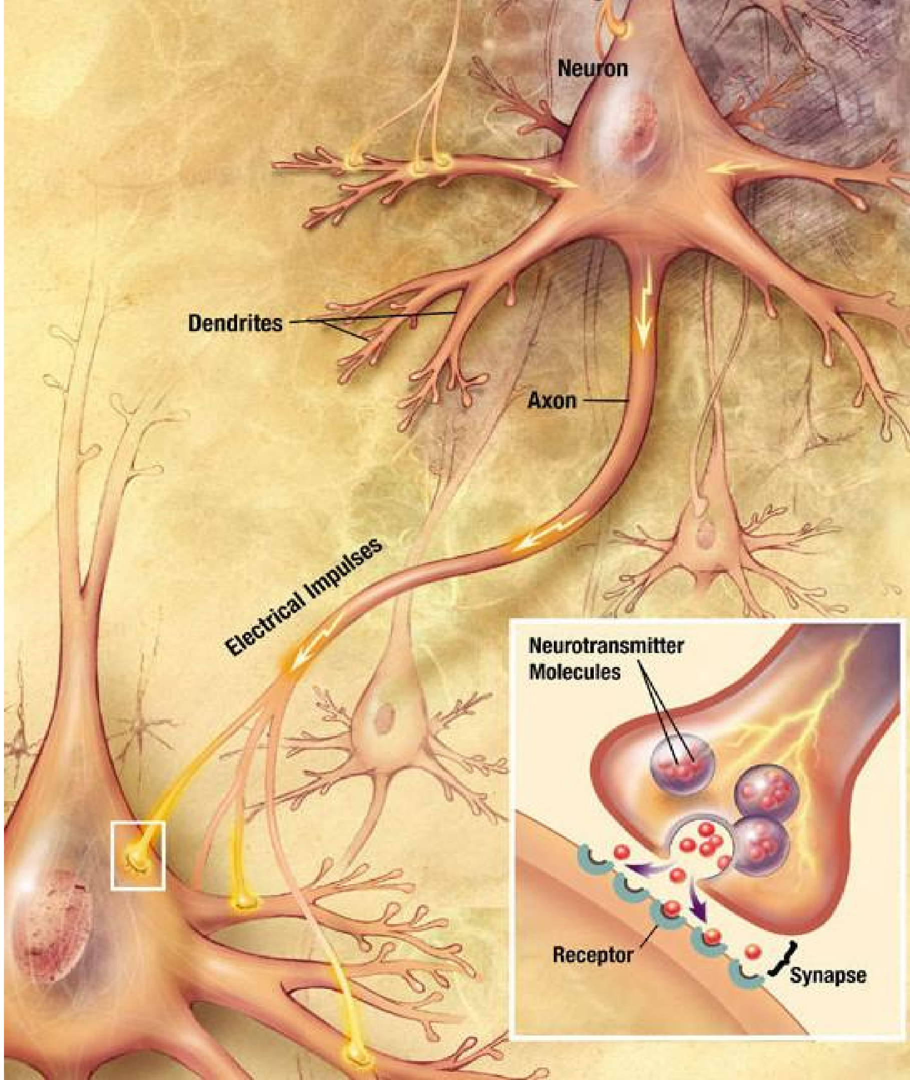
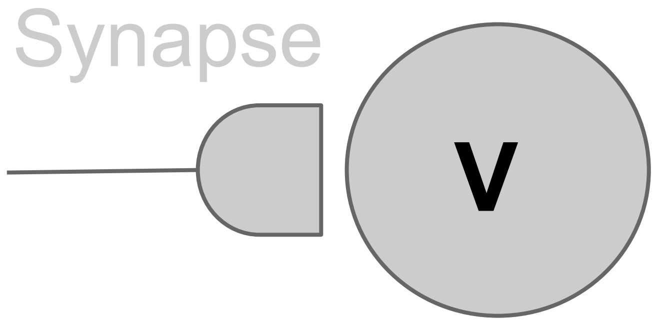


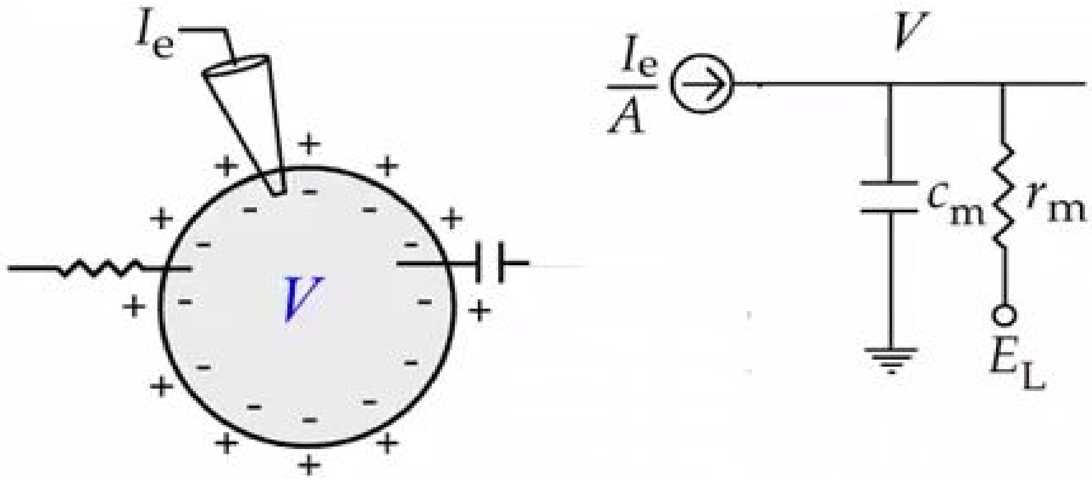
Image source: http://wikimedia.org

Computational Model

We want a computational model of the effects of a synapse on the membrane potential V



RC Circuit Model of Membrane



$$c_m \approx 10\ \mathrm{nF/mm^2}$$

$$r_m \approx 1\ \mathrm{M\Omega\ mm^2}$$

$$C_m = c_m A$$

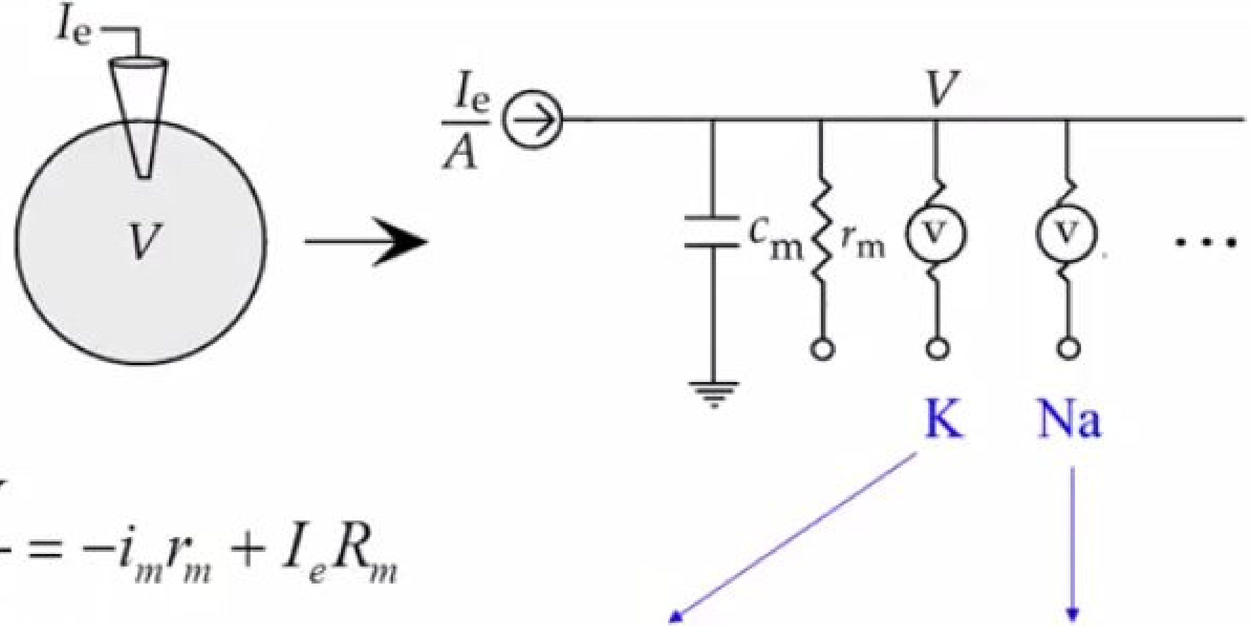
$$R_m = r_m / A$$

$$c_m \frac{dV}{dt} = -\frac{V - E_L}{r_m} + \frac{I_s}{A}$$

Multiply by r_m

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

Learn from Hodgkin and Huxley

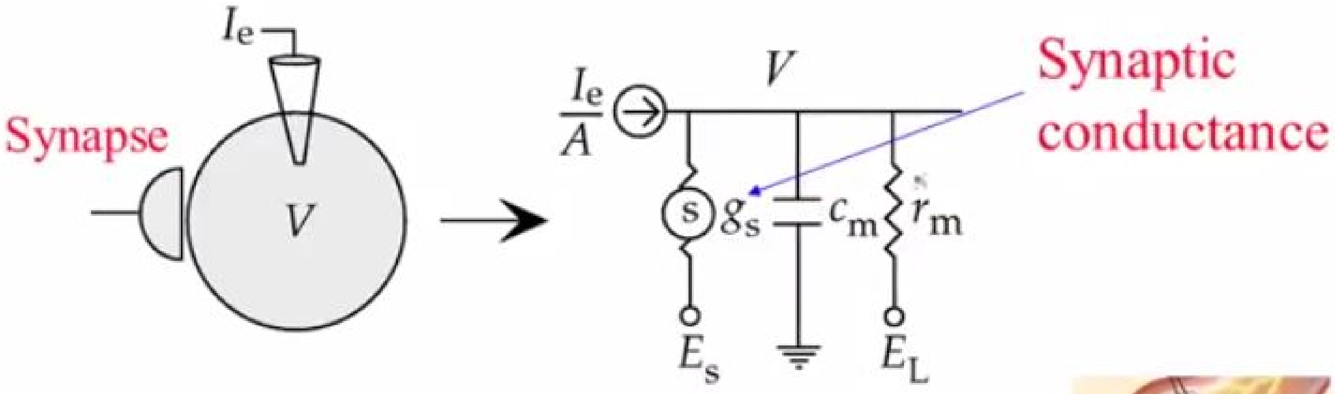


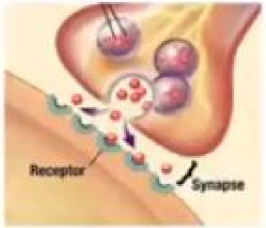
$$\tau_m \frac{dV}{dt} = -i_m r_m + I_e R_m$$

$$i_m = (1/r_m)(V - E_L) + g_{K,\max} n^4 (V - E_K) + g_{Na,\max} m^3 h (V - E_{Na})$$

$E_L = -54$ mV, $E_K = -77$ mV, $E_{Na} = +50$ mV

Membrane Potential with Synapses





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

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

Synaptic Conductance

Assume probability of release is 1...

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

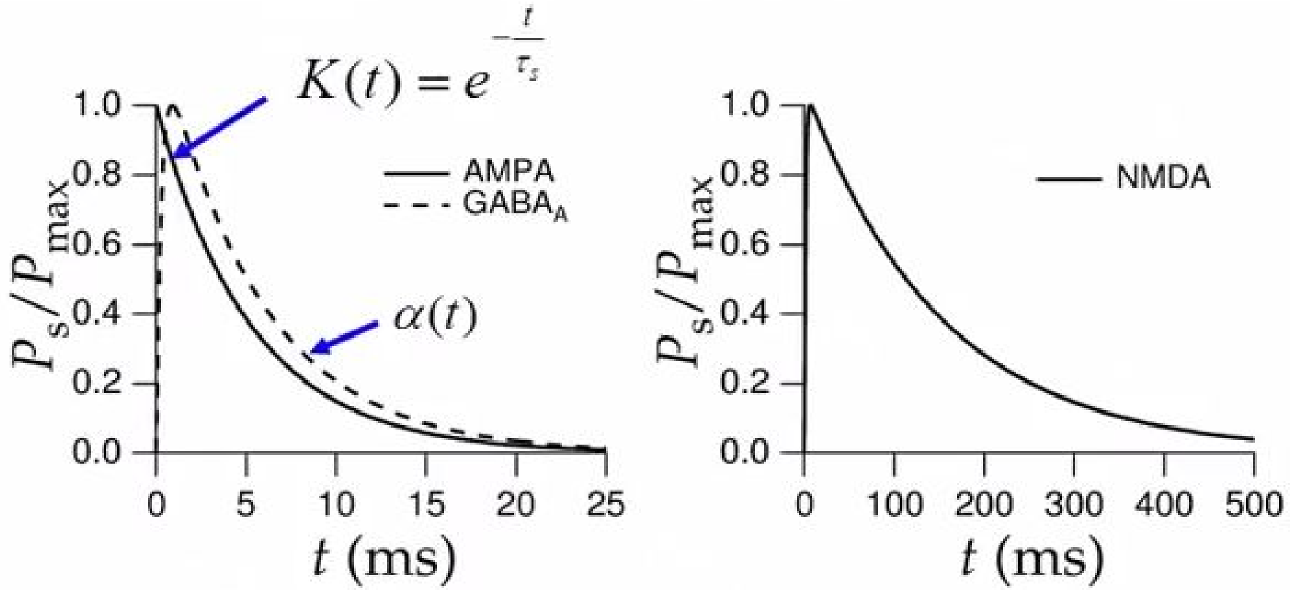
Opening rate

Fraction of channels closed

Closing rate

Fraction of channels open

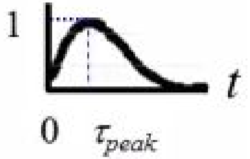
What does Ps look like?



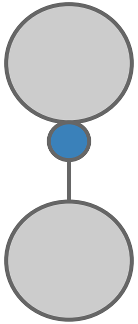
Exponential function gives reasonable fit for some synapses

Others can be fit using "Alpha" function:

$$\alpha(t) = \frac{t}{\tau_{peak}} \cdot e^{(1 - \frac{t}{\tau_{peak}})}$$



Linear Filter Model



Synapse b

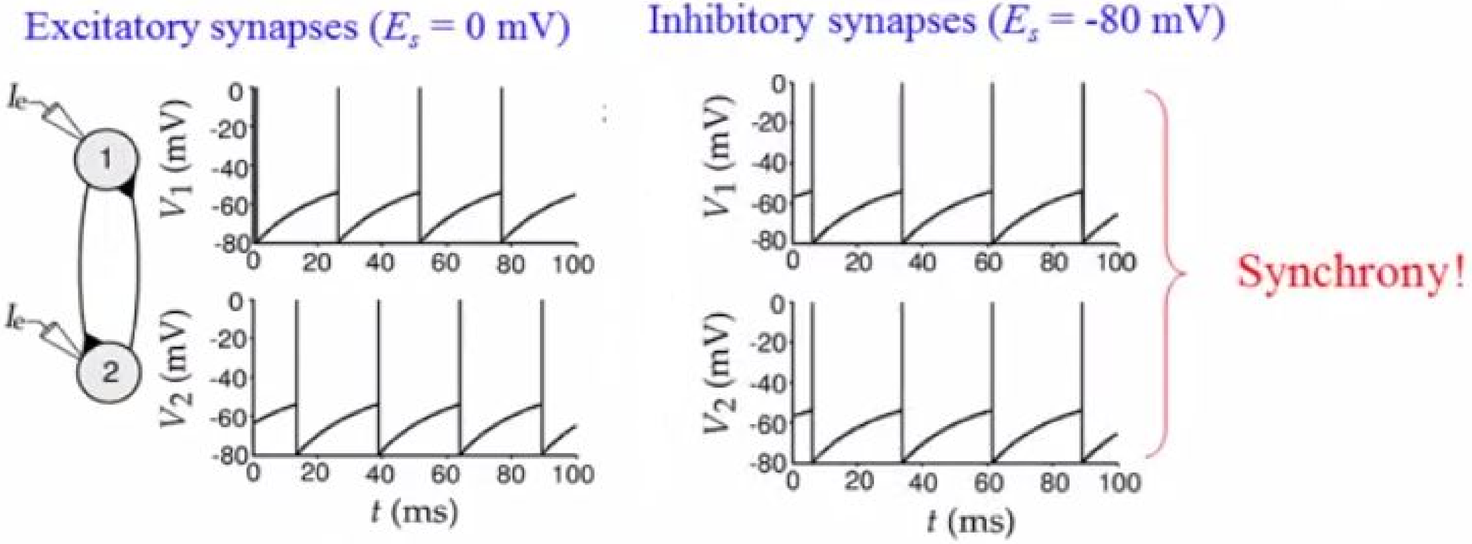
Input spike train

$$\rho_b(t) = \sum_i \delta(t - t_i)$$

Choose a filter for synapse b: K(t)

$$g_b(t) = g_{b,max} \int_{-\infty}^{t} K(t-\tau)\rho_b(\tau)d\tau$$

2-Node Network Example



$$\text{Each neuron:} \quad \tau_m \frac{dV}{dt} = -((V - E_L) - g_s(t)(V - E_s)r_m) + I_e R_m$$

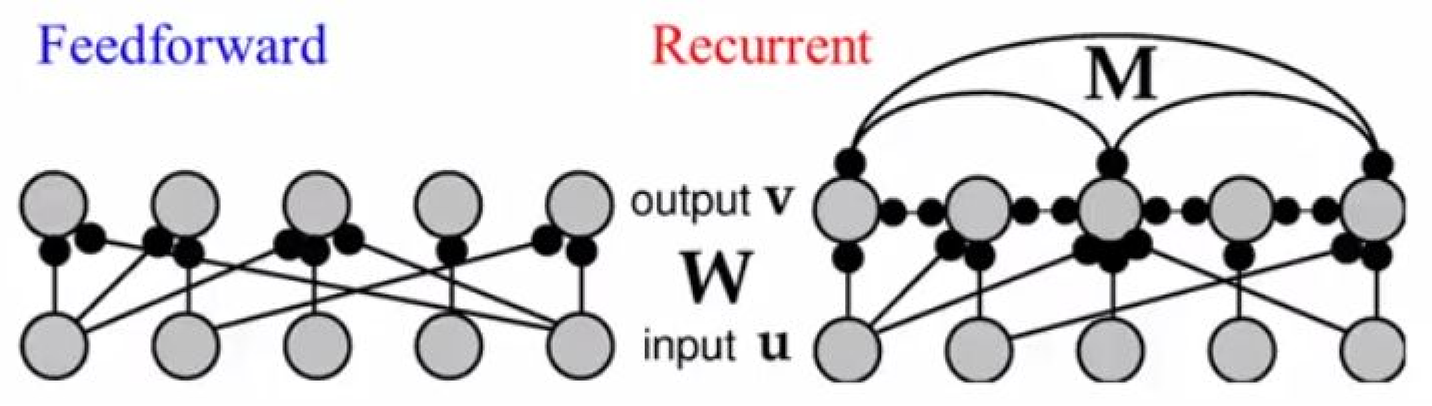
Synapses : Alpha function

$E_L = -70$ mV $\quad V_{thresh} = -54$ mV

$\tau_m = 20$ ms $\quad \tau_{peak} = 10$ ms $\quad I_e R_m = 25$ mV

Modeling Networks of Neurons

Choose between Spiking and Firing Rate Models



Spiking VS Firing Rate

Spiking

Pros: Can model

1. Spike timing
2. Synchrony

Cons:

1. Computationally Expensive

Firing Rate

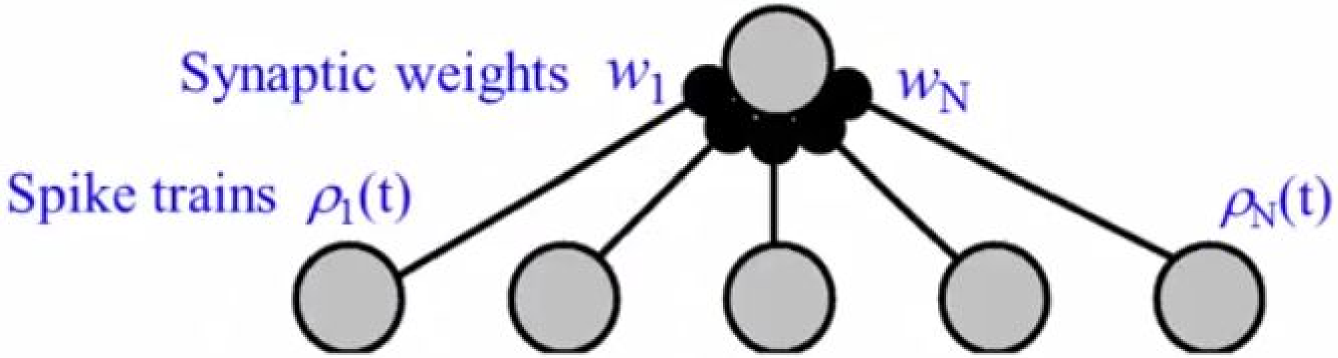
Pros:

1. Computationally Efficient
2. Scalable

Cons:

1. Ignores Spike Timing

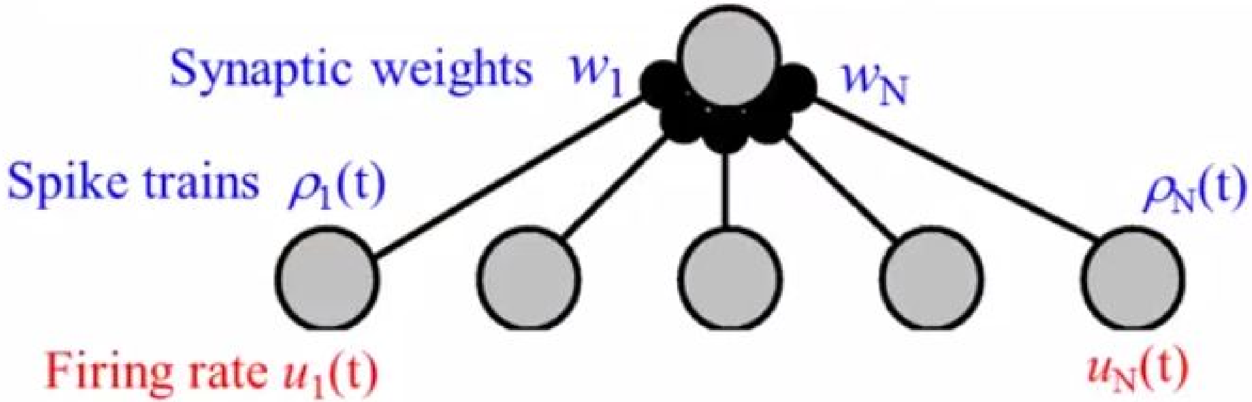
Adding Multiple Synapses



Total synaptic current $I_s(t) = \sum_{b=1}^{N} I_b(t)$

$$I_s(t) = \sum_{b=1}^{N} w_b \int_{-\infty}^{t} K(t-\tau)\rho_b(\tau)d\tau$$

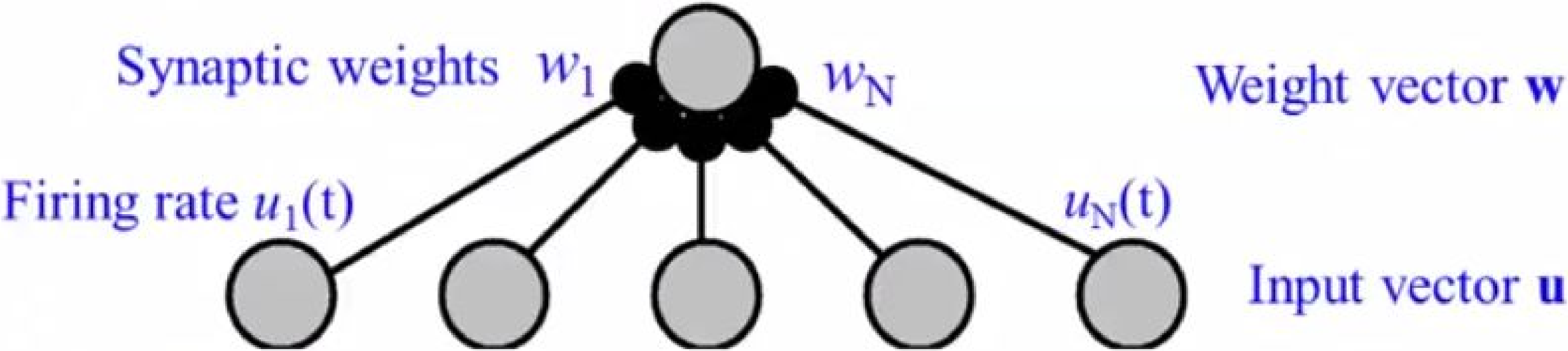
From Spiking to Firing Rate



Total synaptic current

$$I_s(t) = \sum_{b=1}^{N} w_b \int_{-\infty}^{t} K(t-\tau)\rho_b(\tau)d\tau$$ **Spike train $\rho_b(t)$**

$$\approx \sum_{b=1}^{N} w_b \int_{-\infty}^{t} K(t-\tau)u_b(\tau)d\tau$$ **Firing rate $u_b(t)$**



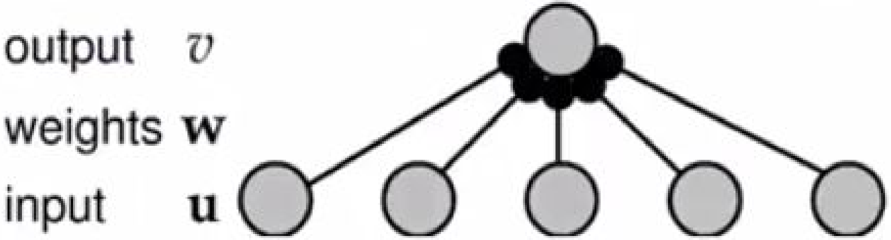
Suppose synaptic filter K is exponential: $K(t) = \frac{1}{\tau_s} e^{-\frac{t}{\tau_s}}$

Differentiating $I_s(t) = \sum_b w_b \int_{-\infty}^{t} K(t-\tau) u_b(\tau) d\tau$ w.r.t. time t,

we get $\tau_s \frac{dI_s}{dt} = -I_s + \sum_b w_b u_b$

$$= -I_s + \mathbf{w} \cdot \mathbf{u}$$

Final Firing-Rate-Based Model



Output firing rate changes like this: $\tau_r \frac{dv}{dt} = -v + F(I_s(t))$

Input current changes like this: $\tau_s \frac{dI_s}{dt} = -I_s + \mathbf{w} \cdot \mathbf{u}$

$$I_s = \boldsymbol{w} \cdot \boldsymbol{u}$$

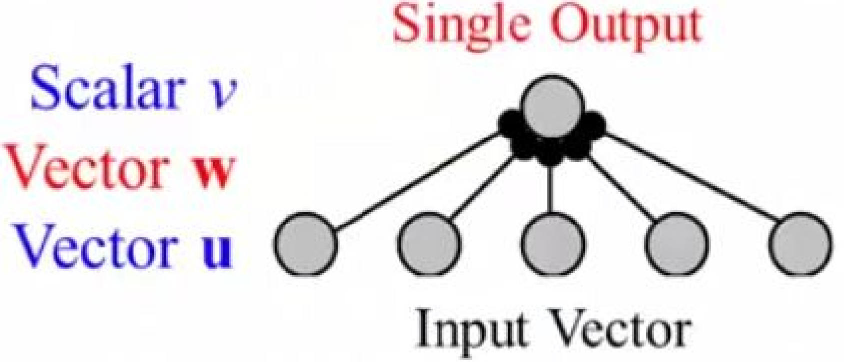
$$\tau_r \frac{dV}{dt} = -V + F(\boldsymbol{w} \cdot \boldsymbol{u})$$

$$V = F(I_s(t))$$

STATIC INPUT

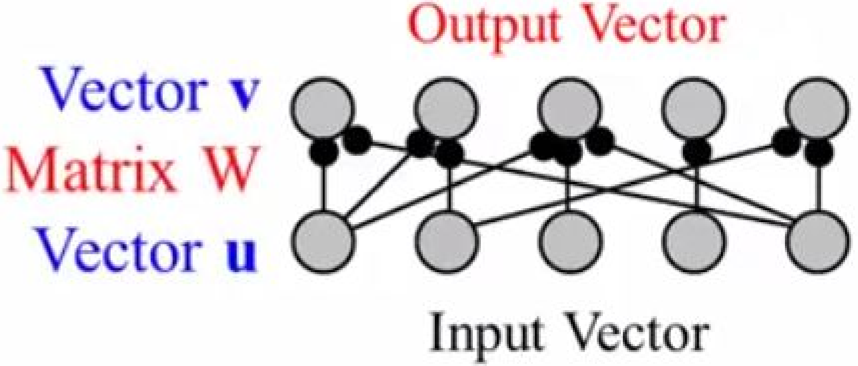
$$V_{ss} = F(\boldsymbol{w} \cdot \boldsymbol{u})$$

What about Multiple Outputs?



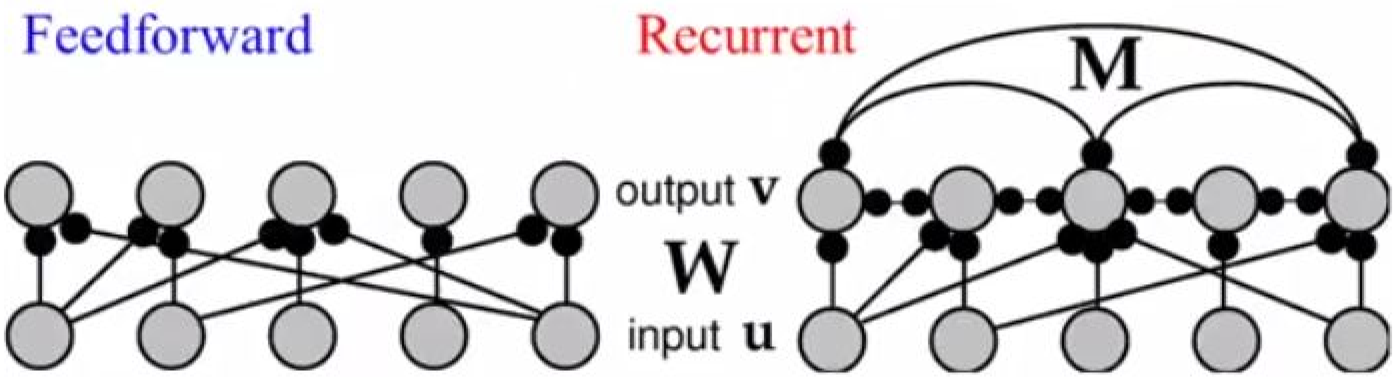
$$\tau \frac{dv}{dt} = -v + F(\mathbf{w} \cdot \mathbf{u})$$

(Assuming relatively fast synapses, $I_s = \mathbf{w} \cdot \mathbf{u}$ at each t)



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

Feedforward vs 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, M = matrix of zeros

Linear Feedforward Network Example

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

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

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

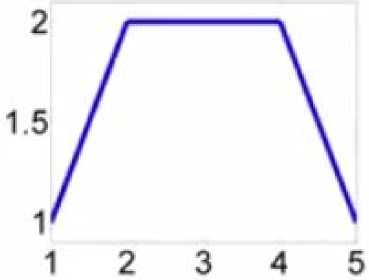
What is the network doing?

Linear Filter: Edge Detection

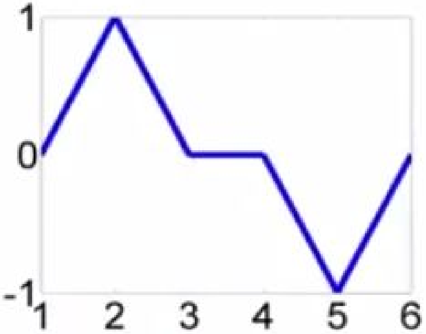
$$\text{Filter} = [0 \quad -1 \quad 1 \quad 0 \quad 0]$$

(and shifted versions in W)

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

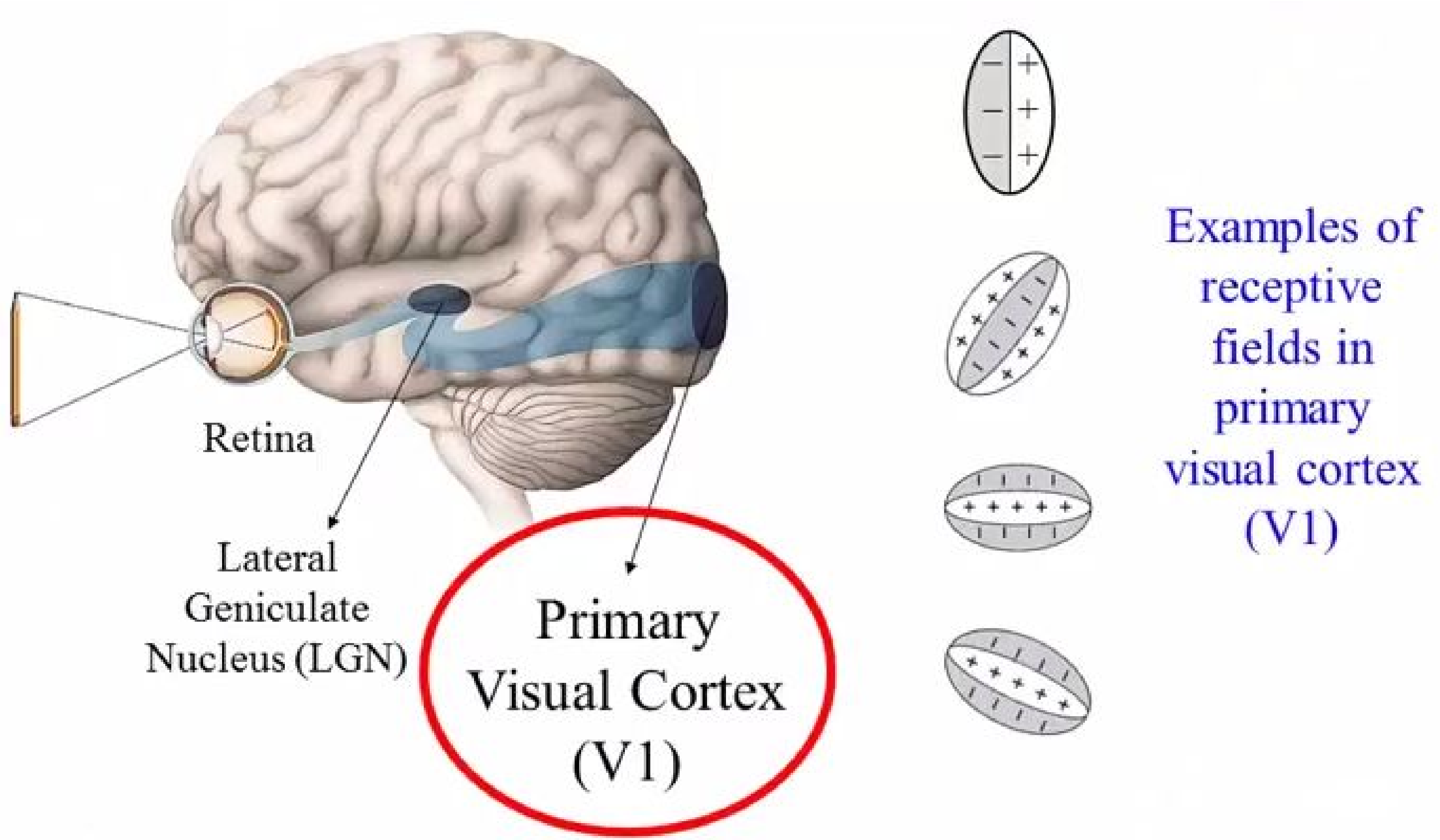


Input



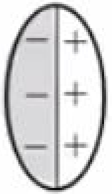
Output

Edge Detectors in the Brain



The Brain Does Calculus

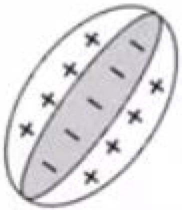
V1 neurons are basically computing derivatives!



$$[0 \quad -1 \quad 1 \quad 0 \quad 0]$$

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

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

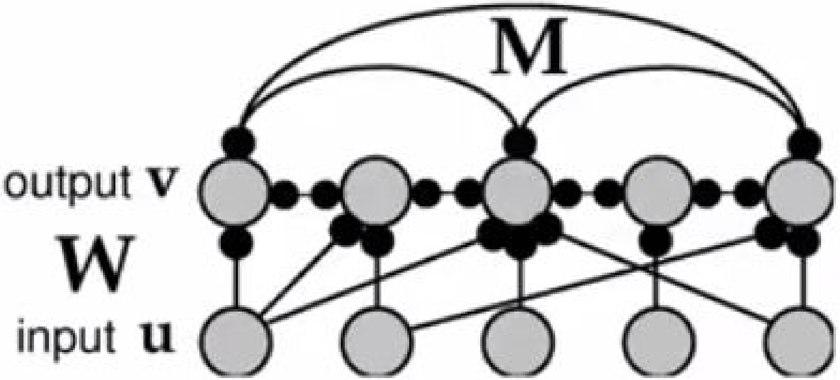


$$[0 \quad 1 \quad -2 \quad 1 \quad 0]$$

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

Disc. approx. $\approx \left(f(x+1) - f(x)\right) - \left(f(x) - f(x-1)\right)$

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



What can a Linear Recurrent Network do?

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

See how v(t) changes as M changes

$$\tau \frac{dv}{dt} = -v + h + Mv$$

- Use eigen vectors of M to solve differential equation for v

- Assume the NxN matrix M is symmetric, then

$$Me_i = \lambda_i e_i$$

- And the solution v can be expressed in an eigenbasis

Use Eigenvectors to solve for v(t)

Substituting $\mathbf{v}(t) = \sum_{i=1}^{N} c_i(t)\mathbf{e}_i$ into differential equation yields

$$\tau \frac{d \sum_{i=1}^{N} c_i(t)\mathbf{e}_i}{dt} = -\sum_{i=1}^{N} c_i(t)\mathbf{e}_i + \mathbf{h} + M \sum_{i=1}^{N} c_i(t)\mathbf{e}_i$$

$$\tau \sum_{i=1}^{N} \frac{dc_i(t)}{dt}\mathbf{e}_i = -\sum_{i=1}^{N} c_i(t)(\mathbf{e}_i - M\mathbf{e}_i) + \mathbf{h}$$

$$\tau \sum_{i=1}^{N} \frac{dc_i(t)}{dt}\mathbf{e}_i = -\sum_{i=1}^{N} c_i(t)(\mathbf{e}_i - \lambda_i \mathbf{e}_i) + \mathbf{h}$$

Substitute using $Me_i = \lambda_i e_i$

Use Eigenvector to solve for v(t) 2

Use orthonormality of eigenbasis and dot both sides with $\mathbf{e}_j$

$$\left(\tau \sum_{i=1}^{N} \frac{dc_i(t)}{dt} \mathbf{e}_i\right) \cdot \mathbf{e}_j = \left(-\sum_{i=1}^{N} c_i(t)(\mathbf{e}_i - \lambda_i \mathbf{e}_i) + \boldsymbol{h}\right) \cdot \mathbf{e}_j$$

$$\tau \frac{dc_j(t)}{dt} = -c_j(t)(1-\lambda_j) + \boldsymbol{h} \cdot \mathbf{e}_j$$

$$c_j(t) = \frac{\boldsymbol{h} \cdot \mathbf{e}_j}{1-\lambda_j}\left[1-\exp\left(\frac{-t(1-\lambda_j)}{\tau}\right) + c_j(0)\exp\left(\frac{-t(1-\lambda_j)}{\tau}\right)\right]$$

Solve first order linear ODE for c(t)

$$\mathbf{v}(t) = \sum_{j=1}^{N} c_j(t)\mathbf{e}_j$$

Eigenvalues determine Network Stability

$$\mathbf{v}(t) = \sum_{j=1}^{N} c_j(t)\mathbf{e}_j \qquad c_j(t) = \frac{\boldsymbol{h}\cdot\mathbf{e}_j}{1-\lambda_j}\left[1-\exp\left(\frac{-t(1-\lambda_j)}{\tau}\right)+c_j(0)\exp\left(\frac{-t(1-\lambda_j)}{\tau}\right)\right]$$

If any $\lambda_j > 1$, then v(t) explodes and the network is unstable

If all $\lambda_j < 1$, then v(t) converges to the steady state solution

$$\mathbf{v}_{ss} = \frac{\mathbf{h}\cdot\mathbf{e}_j}{1-\lambda_j}\cdot\mathbf{e}_j$$

We can now Explore the Network

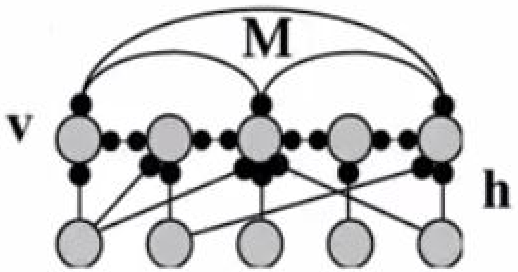
$$\mathbf{v}_{ss} = \sum_i \frac{\mathbf{h} \cdot \mathbf{e}_i}{1 - \lambda_i} \mathbf{e}_i$$

If all $\lambda_i < 1$ and one λ_i (say λ_1) is close to 1 with others much smaller :

$$\mathbf{v}_{ss} \approx \frac{\mathbf{h} \cdot \mathbf{e}_1}{1 - \lambda_1} \mathbf{e}_1$$

Amplification of input projection by a factor of $\frac{1}{1-\lambda_1}$

Linear Recurrent Network Example

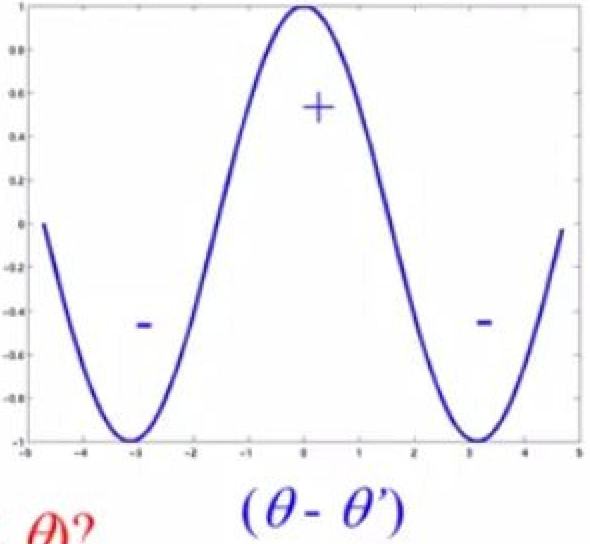


Each output neuron codes for an angle between -180 to +180 degrees

Recurrent connections M = cosine function of relative angle

$$M(\theta,\theta') \propto \cos(\theta - \theta')$$

Excitation nearby, Inhibition further away

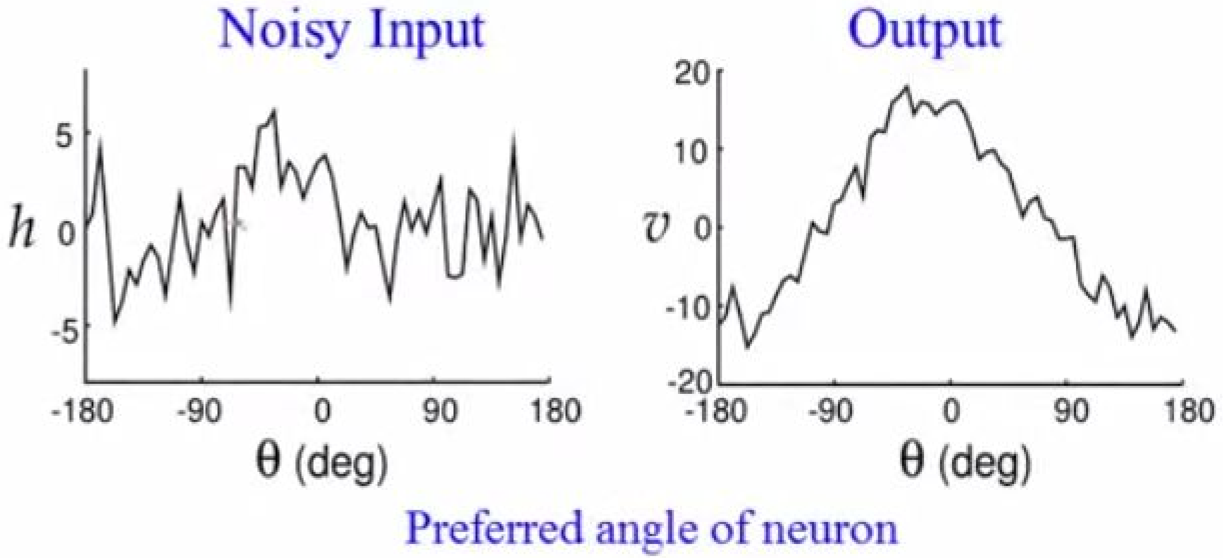


Is M symmetric? $M(\theta, \theta') = M(\theta', \theta)$?

Linear Recurrent Network Amplification

$$M(\theta,\theta') \propto \cos(\theta - \theta'), \text{ all eigenvalues} = 0 \text{ except } \lambda_1 = 0.9$$

$$\text{Amplification } \mathbf{v}_{ss} \approx \frac{(\mathbf{e}_1 \cdot \mathbf{h})\mathbf{e}_1}{1 - \lambda_1} = 10 \times (\mathbf{e}_1 \cdot \mathbf{h})\mathbf{e}_1$$



Linear Recurrent Networks: Memory

$$\tau \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{h} + \mathrm{M}\mathbf{v} \qquad \mathbf{v}(t) = \sum_{i=1}^{N} c_i(t)\mathbf{e}_i$$

Suppose $\lambda_1 = 1$ and all other $\lambda_i < 1$. Then, $\tau \frac{dc_1}{dt} = \mathbf{h} \cdot \mathbf{e}_1$

If input $\mathbf{h}$ is turned on and then off, can show that even after $\mathbf{h} = 0$:

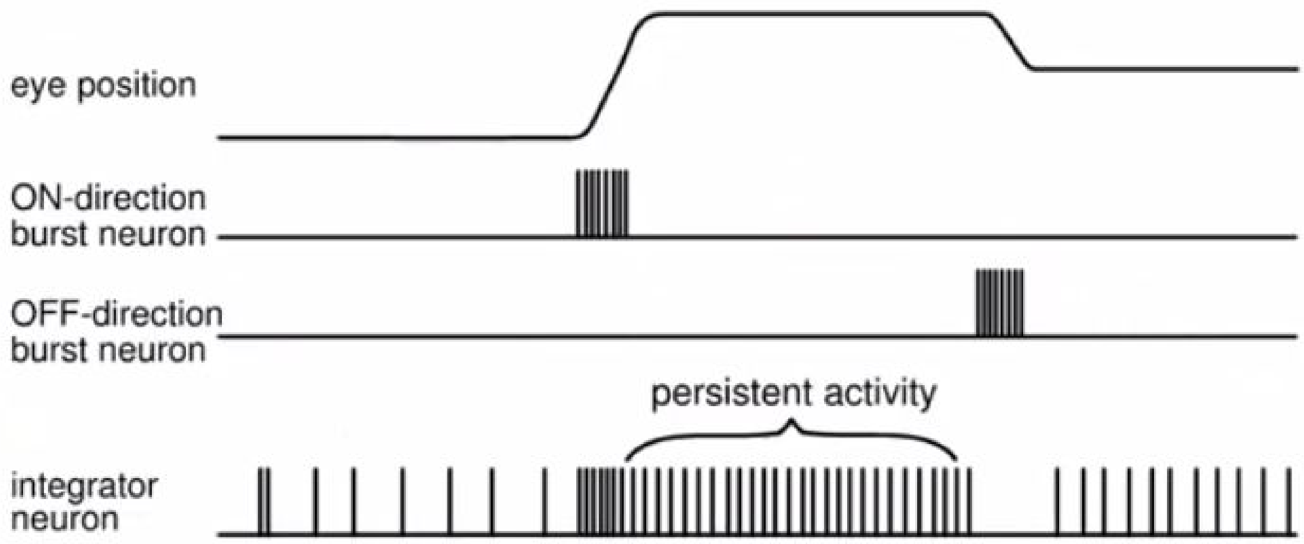
$$\mathbf{v}(t) = \sum_i c_i(t)\mathbf{e}_i$$

$$\approx c_1\mathbf{e}_1 = \frac{\mathbf{e}_1}{\tau}\int_0^t \mathbf{h}(t')\cdot\mathbf{e}_1 dt'$$

Sustained activity without any input!

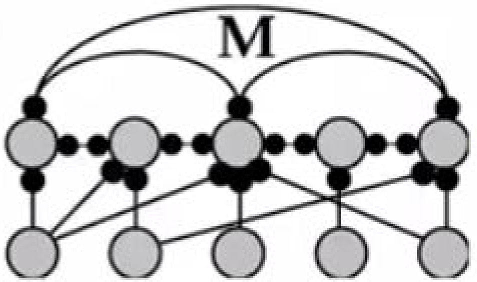
Networks keeps a memory of integral of past input

The Brain can do Calculus: Integration



Input: Bursts of spikes from brain stem oculomotor neurons
Output: Memory of eye position in medial vestibular nucleus

Nonlinear Recurrent Networks

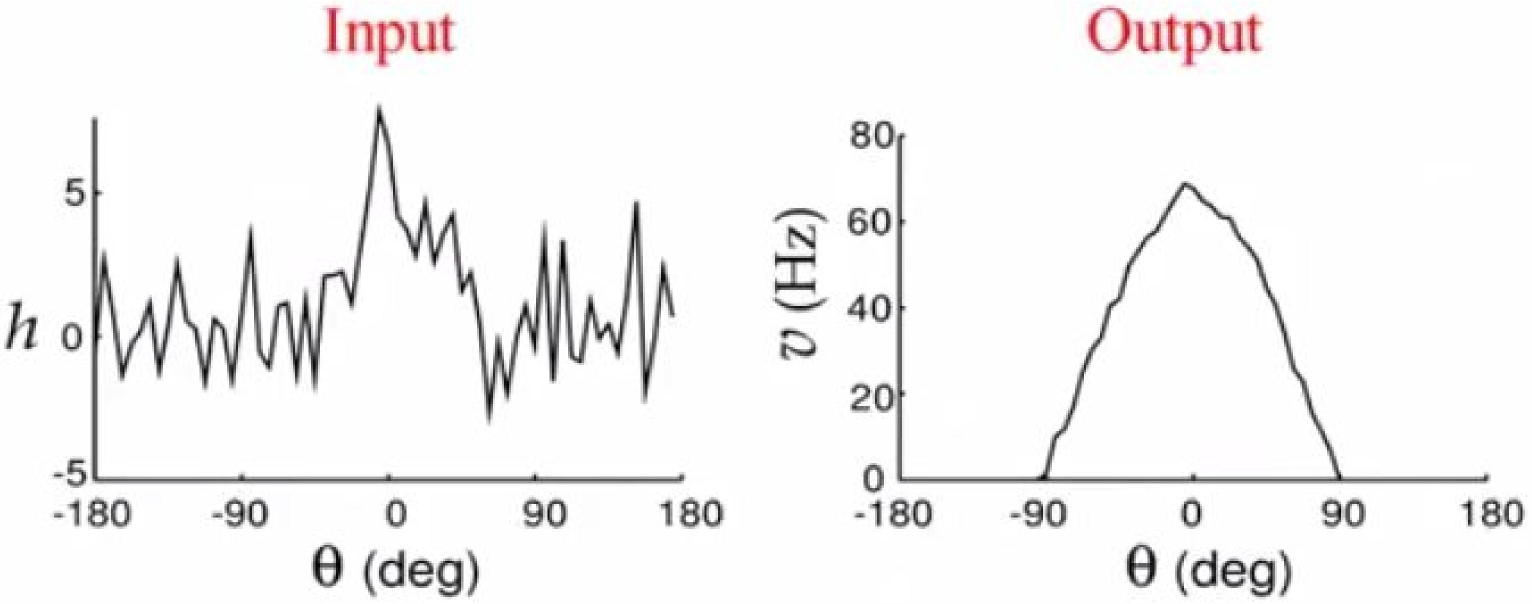


Rectification

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

Output Decay Input Recurrent Feedback

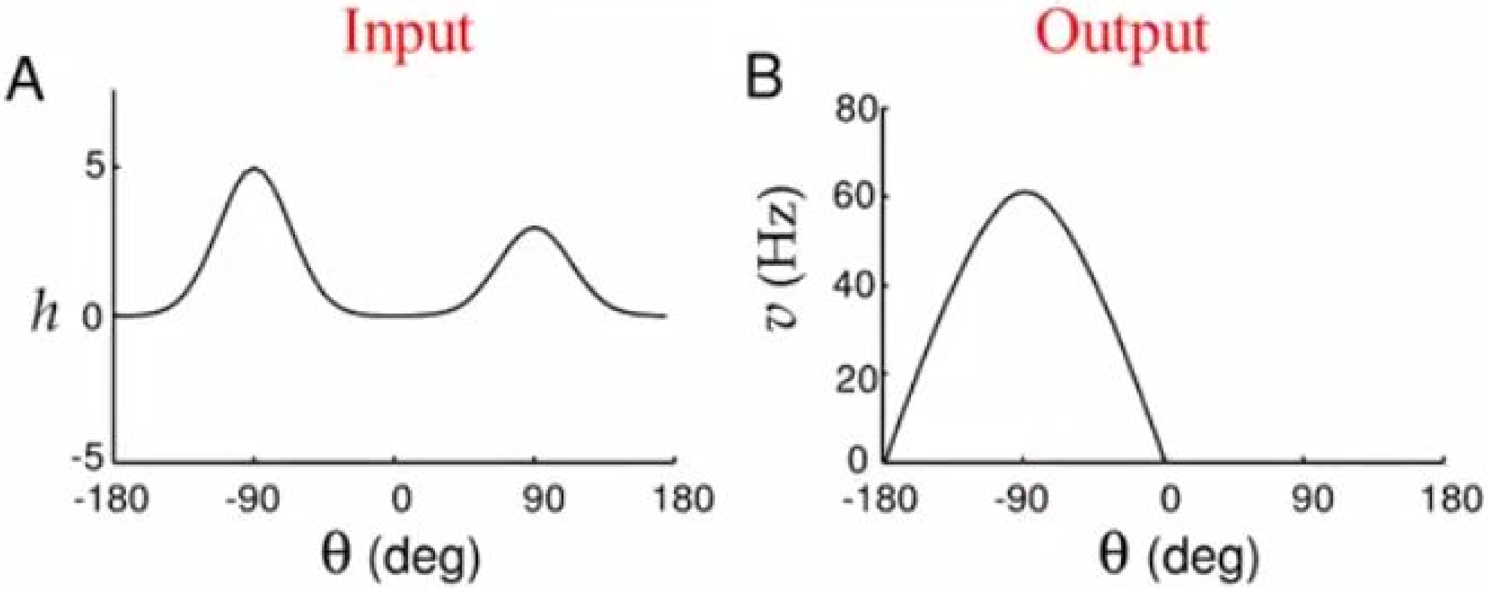
Nonlinear Recurrent Network Performs Amplification



As before, recurrent connections $M(\theta, \theta') \propto \cos(\theta - \theta')$

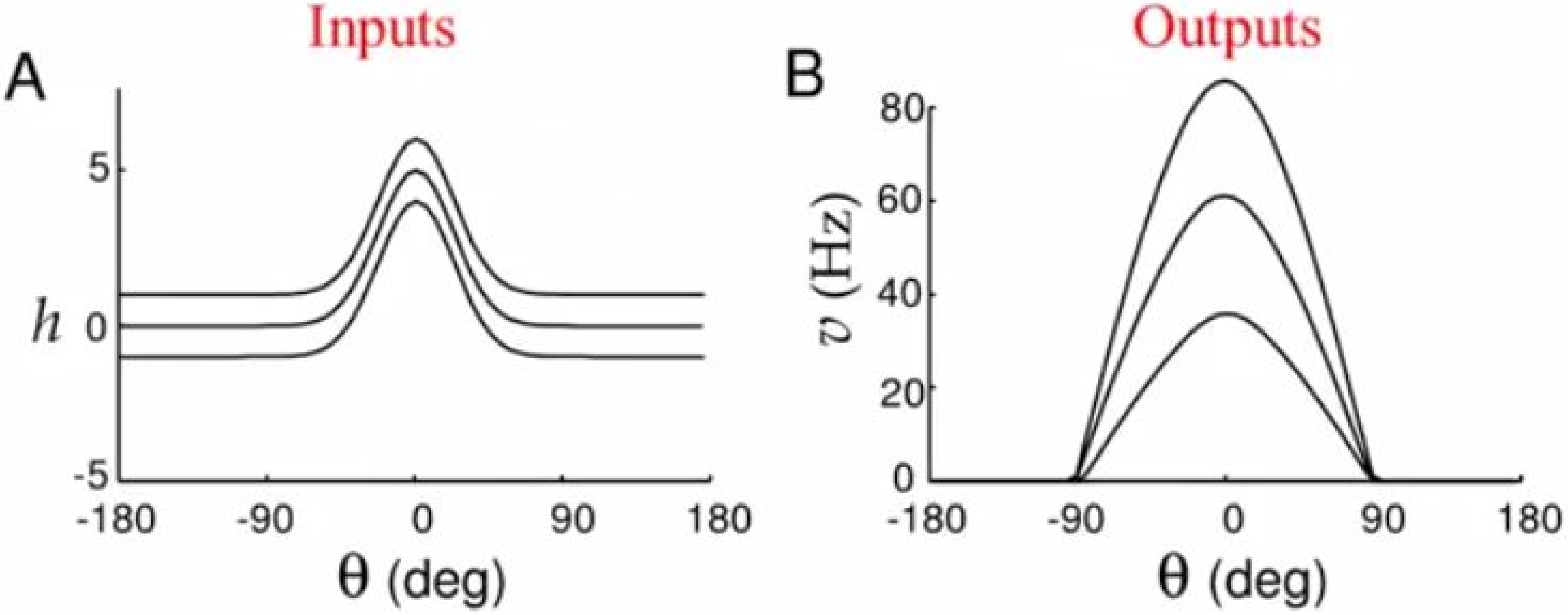
All eigenvalues $= 0$ but $\lambda_1 = 1.9$ (yet stable due to rectification)

Same Nonlinear Network Performs "Selective Attention"



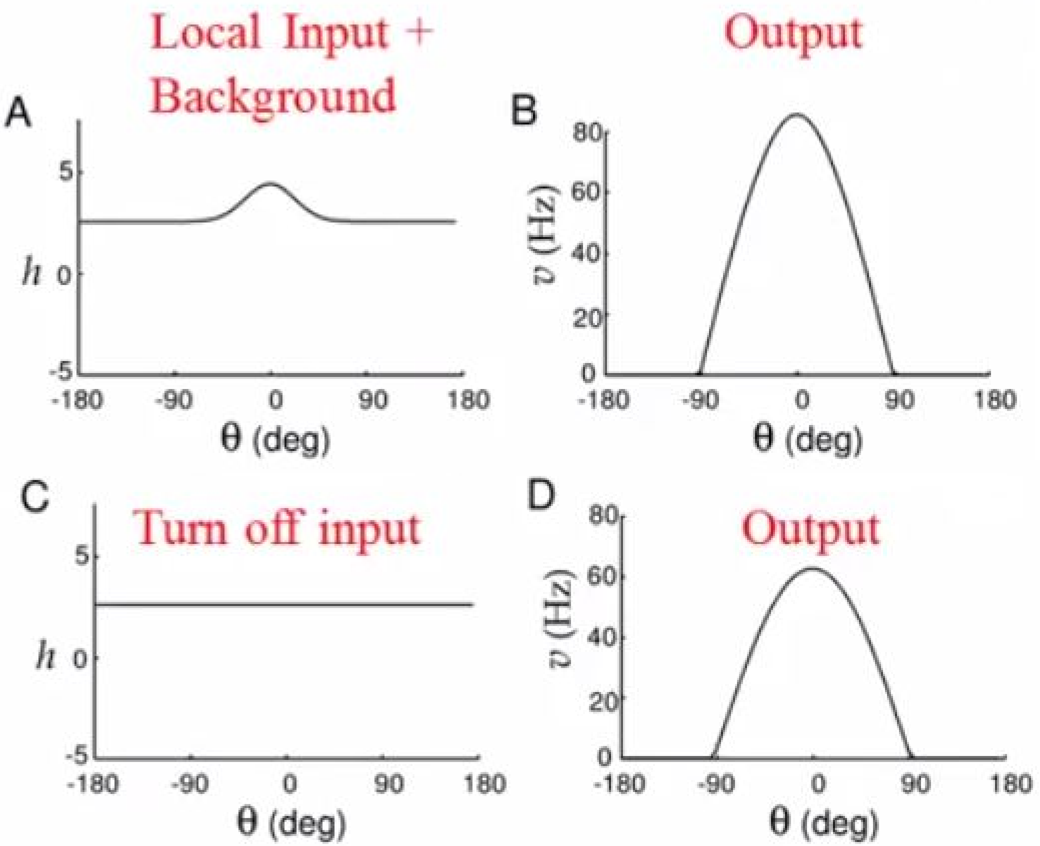
Network performs "Winner-Takes-All" input selection

Nonlinear Network Performs Gain Modulation



Adding a constant amount to the input *h* **multiplies** the output

Memory in Nonlinear Recurrent Networks



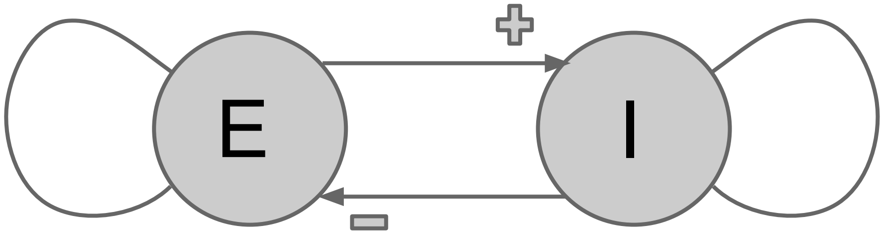
Memory is maintained by recurrent activity

Network maintains a memory of previous activity when input is turned off.

Similar to "short term memory" or "working memory" in prefrontal cortex.

Nonsymmetric Recurrent Networks

Example: Network of Excitatory (E) and Inhibitory (I) neurons



$$\tau_E \frac{dv_E}{dt} = -v_E + [M_{EE} v_E + M_{EI} v_I - \gamma_E]^+$$

$$\tau_I \frac{dv_I}{dt} = -v_I + [M_{II} v_I + M_{IE} v_E - \gamma_I]^+$$

10ms → τ_E; M_{EE} = 1.25, M_{EI} = -1, γ_E = 10

M_{II} = 0, M_{IE} = 1, γ_I = -10

Vary parameter to study the network behavior → τ_I

How do we analyze the dynamics of such a network?

Linear Stability Analysis

$$\frac{dv_E}{dt} = \frac{-v_E + [M_{EE}v_E + M_{EI}v_I - \gamma_I]^+}{\tau_E}$$

$$\frac{dv_I}{dt} = \frac{-v_I + [M_{II}v_I + M_{IE}v_E - \gamma_I]^+}{\tau_I}$$

Take derivative of right hand side with respect to both v_E and v_I

Stability Matrix

$$J = \begin{bmatrix} \frac{M_{EE}-1}{\tau_E} & \frac{M_{EI}}{\tau_E} \\ \frac{M_{IE}}{\tau_I} & \frac{M_{II}-1}{\tau_I} \end{bmatrix}$$

- Eigenvalues of J have real and imaginary parts
- These eigenvalues determine dynamics of the nonlinear network near a fixed point

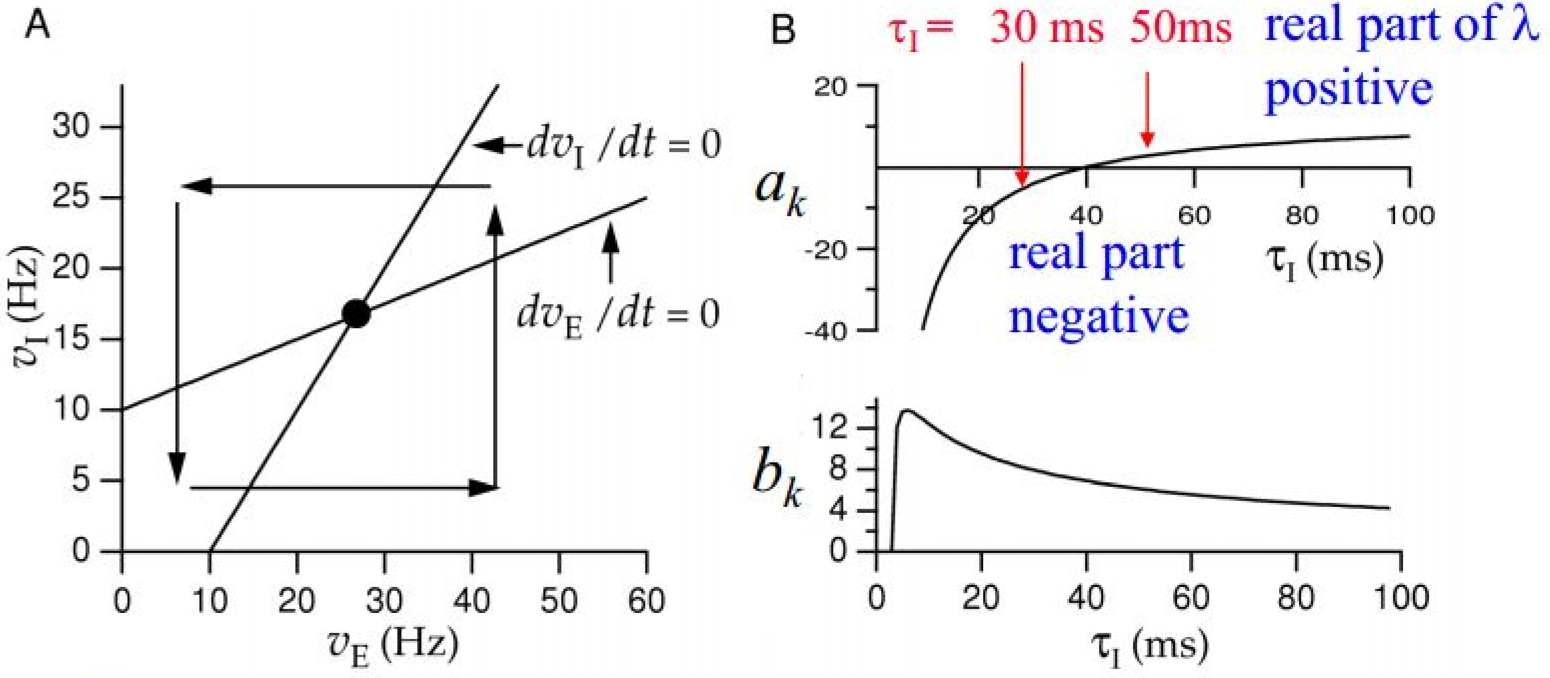
Jacobian Matrix:

$$J = \begin{bmatrix} \dfrac{(M_{EE} - 1)}{\tau_E} & \dfrac{M_{EI}}{\tau_E} \\ \dfrac{M_{IE}}{\tau_I} & \dfrac{(M_{II} - 1)}{\tau_I} \end{bmatrix}$$

Its two eigenvalues (obtained by solving $det(J - \lambda \mathrm{I}) = 0$):

$$\lambda = \frac{1}{2}\left(\frac{(\overset{1.25}{M_{EE}} - 1)}{\underset{10\text{ ms}}{\tau_E}} + \frac{(\overset{0}{M_{II}} - 1)}{\tau_I} \pm \sqrt{\left(\frac{M_{EE} - 1}{\tau_E} - \frac{M_{II} - 1}{\tau_I} \right)^2 + 4 \frac{\overset{-1}{M_{EI}} \overset{1}{M_{IE}}}{\tau_E \tau_I}} \right)$$

Next page plots real and imaginary parts of λ as a function of τ_I



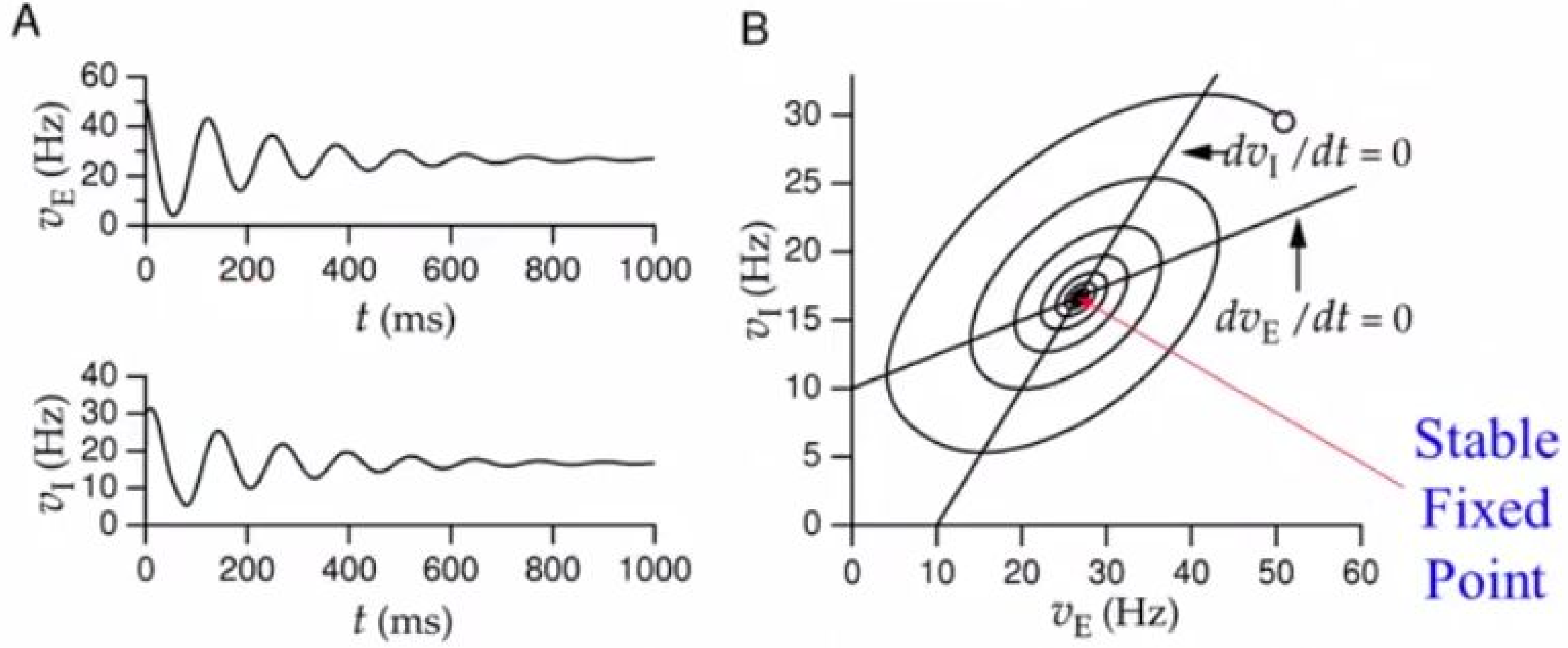
$$10\frac{dv_E}{dt} = -v_E + [1.25v_E - v_I + 10]^+$$

$$\tau_I \frac{dv_I}{dt} = -v_I + [0 \cdot v_I + v_E - 10]^+$$

Real and imaginary parts (a_k and b_k) of λ $(= a_k + ib_k)$ as a function of τ_I

Damped Oscillations in Network

Choose $\tau_I = 30$ ms (makes real part of eigenvalues negative)



Instability and the Limit Cycle

Choose $\tau_I = 50$ ms (makes real part of eigenvalues positive)