CSE/NB 528

Lecture 10: Recurrent Networks

(Chapter 7)

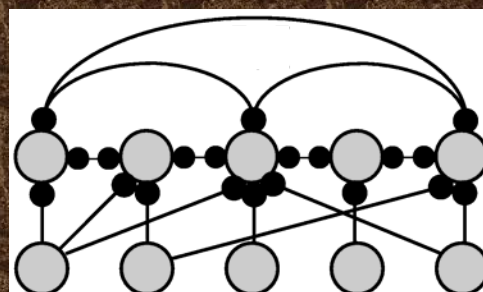


Image from http://clasdean.la.asu.edu/news/images/ubep2001/neuron3.jpg

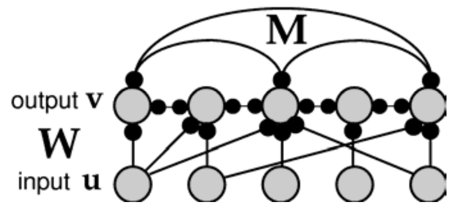
Lecture figures are from Dayan & Abbott's book

http://people.brandeis.edu/~abbott/book/index.html

What's on our platter today?

- Computation in Recurrent Networks
  - Linear Recurrent Networks
    - Stability analysis using eigenvalues
  - Nonlinear Recurrent Networks
    - Can amplify inputs
    - Can select inputs
    - Can multiply (gain modulation)
    - Can store short-term memory
  - Associative Memory (Hopfield net)
    - Showing Stability via Lyapunov function

Recurrent Networks



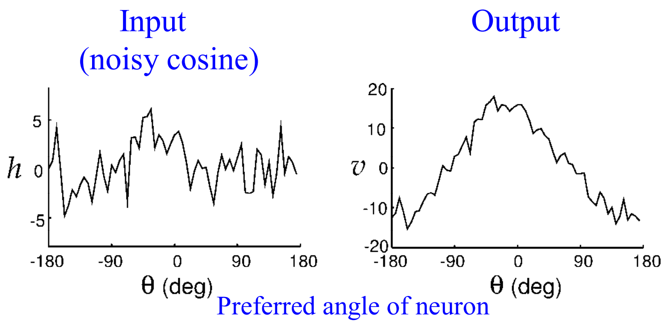
$$\tau \frac{d\mathbf{v}}{dt} = -\mathbf{v} + F(\mathrm{W}\mathbf{u} + \mathrm{M}\mathbf{v})$$

Output Decay Input Feedback

What can a Linear Recurrent Network do?

Analysis based on eigenvectors of recurrent weight matrix

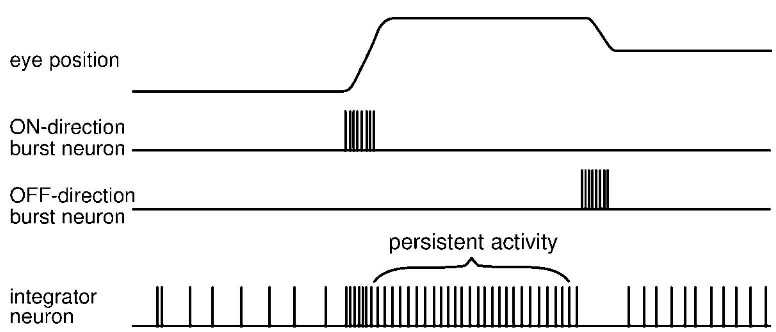
Amplification in a Linear Recurrent Network



$$M(\theta,\theta') = \frac{\lambda_1}{\pi}\cos(\theta-\theta')$$

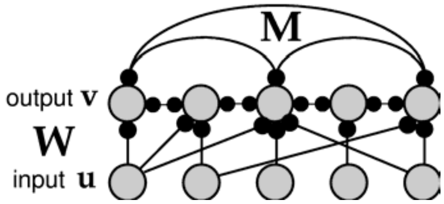
All eigenvalues = 0 except $\lambda_1 = 0.9$ i.e. amplification $= \frac{1}{1-\lambda_1} = 10$

Input Integration for Maintaining Eye Position



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

Nonlinear Recurrent Networks



Two types of firing-rate models

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

Current Dynamics (firing rate v = F(I))

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

Firing-Rate Dynamics

Output Decay Input Feedback

(Convenient to use W**u** = **h**)

Continuous Nonlinear Recurrent Networks

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

$$\tau \frac{dv_i}{dt} = -v_i + F(h_i + \sum_j M_{ij} v_j)$$

Discrete case (small number of neurons)

Continuous case (in the limit of large numbers of neurons):

$$\tau \frac{dv(\theta)}{dt} = -v(\theta) + F(h(\theta) + \rho_\theta \int_{-\pi}^{\pi} M(\theta, \theta') v(\theta') d\theta')$$

θ = preferred stimulus of the neuron (e.g. orientation of input)

Example of a Continuous Recurrent Network

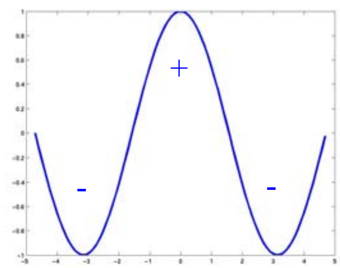
Choose *F* = rectification nonlinearity:

$$\tau \frac{dv(\theta)}{dt} = -v(\theta) + \left[h(\theta) + \int_{-\pi}^{\pi} M(\theta,\theta')v(\theta')d\theta' \right]^{+}$$

$$M(\theta,\theta') = \frac{\lambda_1}{\pi}\cos(\theta - \theta')$$

Choose recurrent connections = cosine function of relative angle

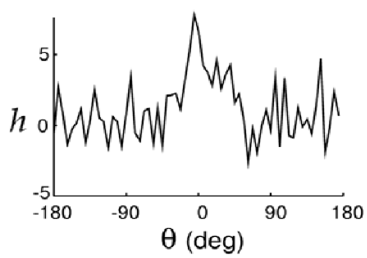
Excitation nearby,
Inhibition further away



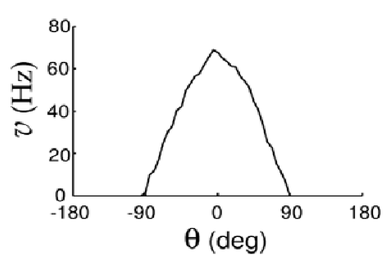
$(\theta - \theta')$

Amplification in a Nonlinear Recurrent Network

Input



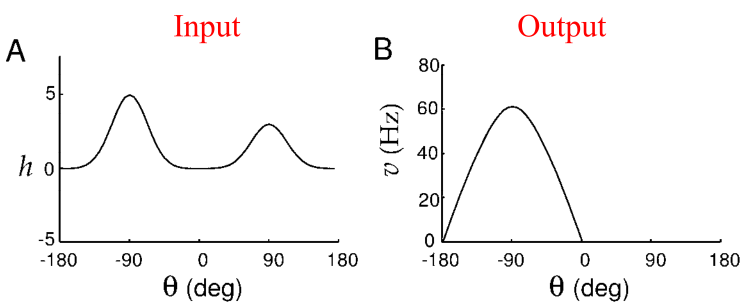
Output



$$\tau \frac{dv(\theta)}{dt} = -v(\theta) + \left[h(\theta) + \frac{\lambda_1}{\pi}\int_{-\pi}^{\pi} \cos(\theta - \theta')v(\theta')d\theta' \right]^{+}$$

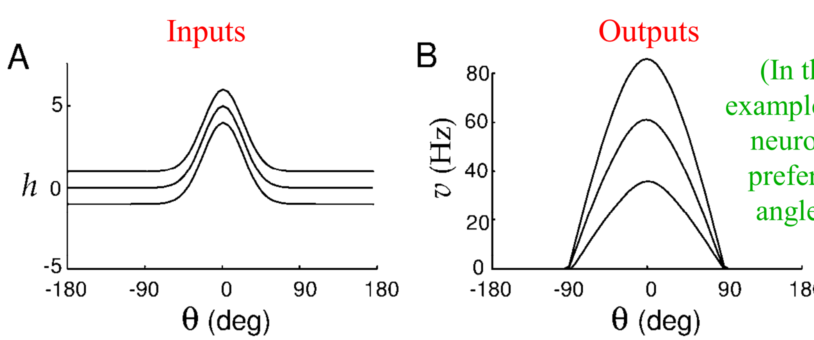
$\lambda_1 = 1.9$ (but stable due to rectification)

Selective "Attention" in a Nonlinear Recurrent Network



Network performs "winner-takes-all" input selection

Gain Modulation in a Nonlinear Recurrent Network



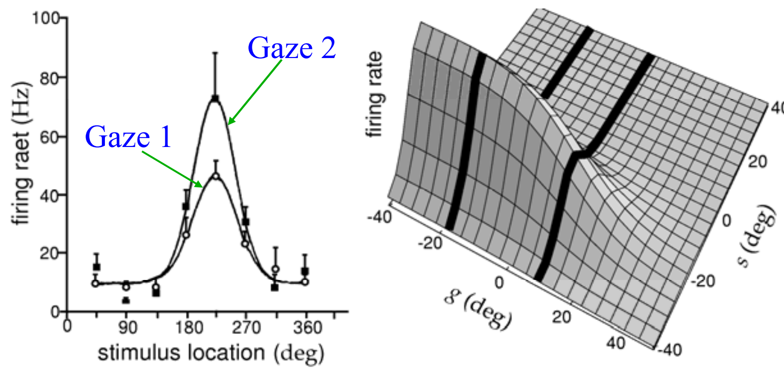
(In this example, θ = neuron's preferred angle s)

Changing the level of input by adding g multiplies the output

If h = s + g (s = stimulus angle on retina, g = gaze angle), then network output is gain-modulated similar to parietal cortex neurons

(Width determined by recurrent matrix *M*)

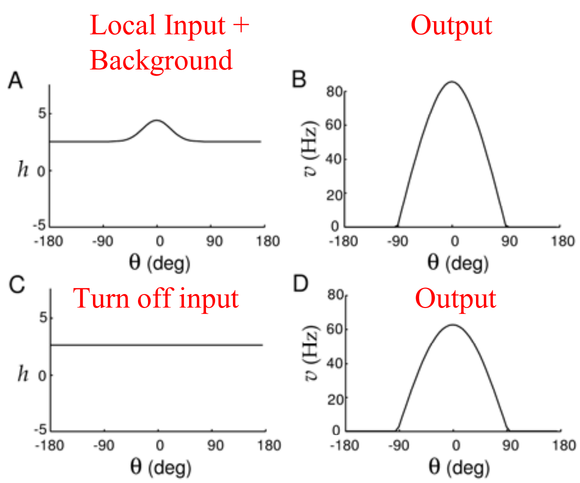
Gain Modulation in Parietal Cortex Neurons



Responses of Area 7a neuron

Example of a gain-modulated tuning curve

Short-Term Memory Storage in a Nonlinear Recurrent Network



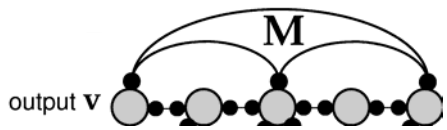
Network maintains a *memory of previous activity* when input is taken away.

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

Memory is maintained by recurrent activity

Associative Memories (Hopfield Networks)

- Fully connected, no feedforward inputs



$$\tau \frac{d\mathbf{I}}{dt} = -\mathbf{I} + \mathrm{M} \cdot g(\mathbf{I}) \quad \text{or,}$$

$$\tau \frac{dI_i}{dt} = -I_i + \sum_j M_{ij} v_j \quad \text{where } v_j = g(I_j)$$

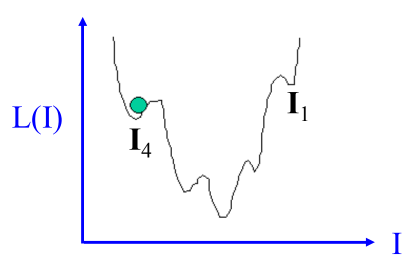
g = sigmoid function

Idea: Store patterns as *fixed points* of this network

Question: Will **I** always converge to a fixed point?

Enter…Lyapunov Functions

- Idea: If dI/dt causes some function L(I) to always decrease or remain constant (i.e. dL/dt ≤ 0) and L has a lower bound (with dL/dt = 0 only if dI/dt = 0), *then dI/dt = 0 eventually*
  - ➩ **Network converges to a fixed point**
- L also called “energy” function or “cost” function



Lyapunov for Hopfield networks

- What is a good Lyapunov function L(I) for Hopfield nets?
- What constraints are required on the recurrent weights **M**?
- On-board example: L(I)

Next Class: Wrap up of network models

- Things to do:
  - Start reading Chapter 8
  - Homework #3 due Tuesday
  - Start on mini-project