CSE/NB 528

Lecture 11: More on Networks

(Chapters 7 & 8)

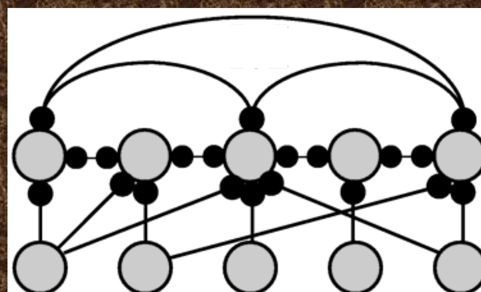


Image from http://clasdean.la.asu.edu/news/images/ubep2001/neuron3.jpg

Lecture figures are from Dayan & Abbott's book
http://people.brandeis.edu/~abbott/book/index.html

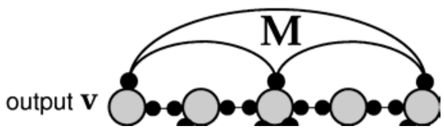
Gameplan for Today

(Copyright, Warner Brothers)

- Wrap up of Nonlinear Recurrent Networks
- Plasticity and Learning
  - Types: Unsupervised, Supervised, and Reinforcement learning
- Unsupervised Learning
  - Hebb rule and its variants (Covariance, BCM, Oja rule)
  - Principal Component Analysis (PCA)
  - Temporally Asymmetric Hebbian learning

Associative Memories (Hopfield Networks)

- Fully connected, no feedforward inputs



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

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

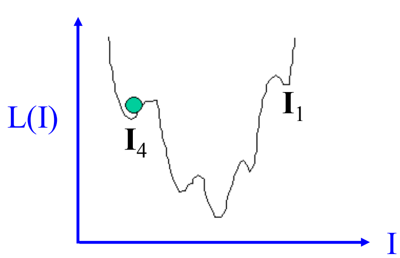
g = sigmoid function

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**?

Lyapunov for Hopfield networks

Given: $\tau \frac{dI_i}{dt} = -I_i + \sum_j M_{ij} v_j$ where $v_j = g(I_j) = \tanh(\beta I_j)$

Define: $L(I) = -\frac{1}{2}\sum_{ij} M_{ij} v_i v_j + \sum_i \int_0^{v_i} g^{-1}(v)dv$

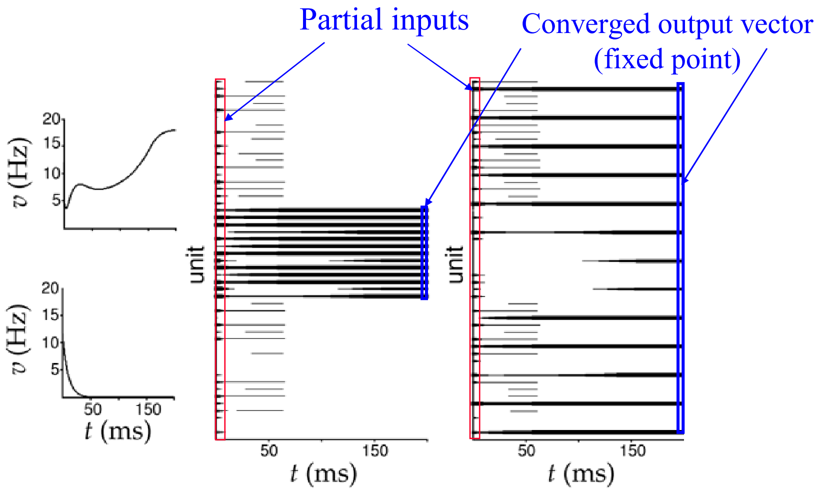
If M is symmetric ($M_{ij} = M_{ji}$), we can show:

$$\frac{dL}{dt} = -\tau \sum_i g'(I_i)\left(\frac{dI_i}{dt}\right)^2 \leq 0$$

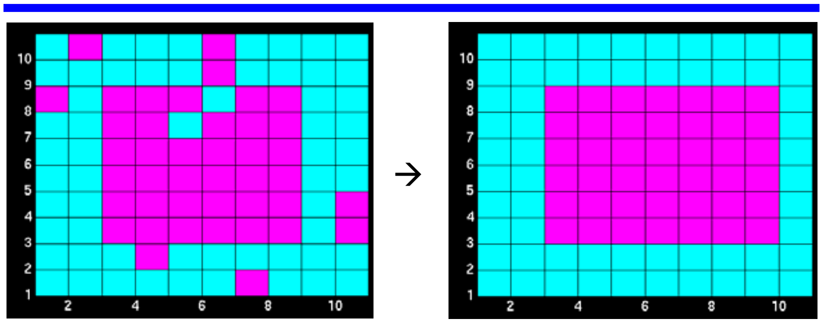
Take-home exercise!

Since L is bounded from below and $dL/dt = 0$ only if $dI_i/dt = 0$, L cannot decrease forever and $dI_i/dt = 0$ eventually for all i

Example of Auto-Associative Memory



Pattern Completion in a Hopfield Network

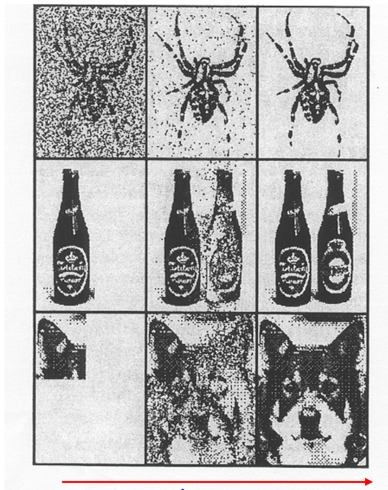


Network converges from here to here

Local minimum ("attractor") of cost (or "energy") function stores pattern

Pattern Recall in Hopfield Nets

Initial states



Stable states (fixed points)

time

What about Non-Symmetric Recurrent Networks?

- Example: Network of Excitatory (E) and Inhibitory (I) Neurons
  - Connections can't be symmetric: Why?

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

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

Simple 2 neuron model for representing interacting populations
One excitatory neuron and one inhibitory neuron

Stability Analysis of Nonlinear Recurrent Networks

General case: $\frac{d\mathbf{v}}{dt} = \mathbf{f}(\mathbf{v})$

Suppose $\mathbf{v}_\infty$ is a fixed point (i.e., $\mathbf{f}(\mathbf{v}_\infty) = 0$)

Near $\mathbf{v}_\infty$, $\mathbf{v}(t) = \mathbf{v}_\infty + \boldsymbol{\varepsilon}(t)$ (i.e., $\frac{d\mathbf{v}}{dt} = \frac{d\boldsymbol{\varepsilon}}{dt}$)

Taylor expansion: $\mathbf{f}(\mathbf{v}(t)) = \mathbf{f}(\mathbf{v}_\infty) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{v}}\right|_{\mathbf{v}_\infty} \boldsymbol{\varepsilon}(t)$

$$i.e.\ \frac{d\mathbf{v}}{dt} = \left.\frac{\partial \mathbf{f}}{\partial \mathbf{v}}\right|_{\mathbf{v}_\infty} \boldsymbol{\varepsilon}(t) = J \cdot \boldsymbol{\varepsilon}(t) = \frac{d\boldsymbol{\varepsilon}}{dt}$$

J is the "Jacobian matrix"

Derive solution for $\mathbf{v}(t)$ based on eigen-analysis of *J*
Eigenvalues of *J* determine stability of network

(see Mathematical Appendix A.3 in textbook)

Example: Non-Symmetric Recurrent Networks

- Specific Network of Excitatory (E) and Inhibitory (I) Neurons:

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

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

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

Parameter → τ_I; 0 → M_{II}; 1 → M_{IE}; 10 → γ_I

Linear Stability Analysis

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

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

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

- Matrix of derivatives (the "Jacobian Matrix"):

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

Compute the Eigenvalues

- Jacobian Matrix:

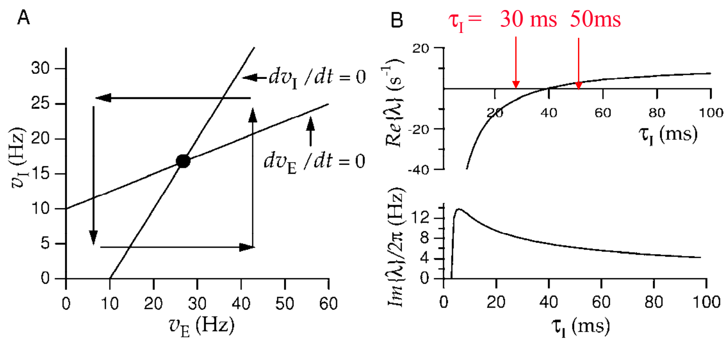
$$J = \begin{bmatrix} \frac{(M_{EE} - 1)}{\tau_E} & \frac{M_{EI}}{\tau_E} \\ \frac{M_{IE}}{\tau_I} & \frac{(M_{II} - 1)}{\tau_I} \end{bmatrix}$$

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

$$\lambda = \frac{1}{2}\left(\frac{(M_{EE} - 1)}{\tau_E} + \frac{(M_{II} - 1)}{\tau_I} \pm \sqrt{\left(\frac{M_{EE} - 1}{\tau_E} - \frac{M_{II} - 1}{\tau_I} \right)^2 + 4 \frac{M_{EI} M_{IE}}{\tau_E \tau_I}} \right)$$

Different dynamics depending on real and imaginary parts of λ (see pages 410-412 of Appendix in Text)

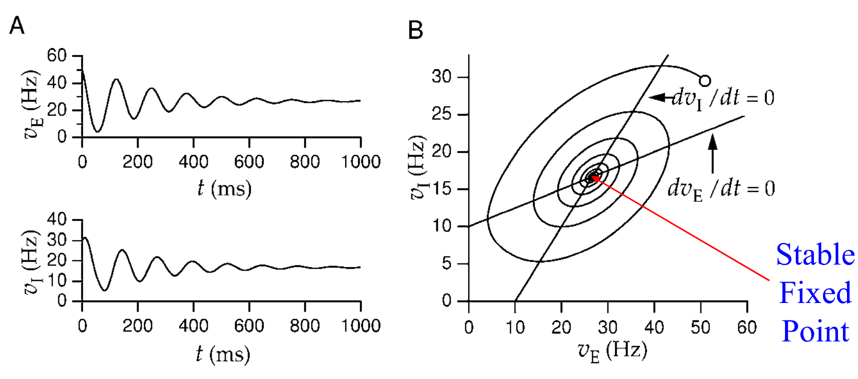
Phase Plane and Eigenvalue Analysis



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

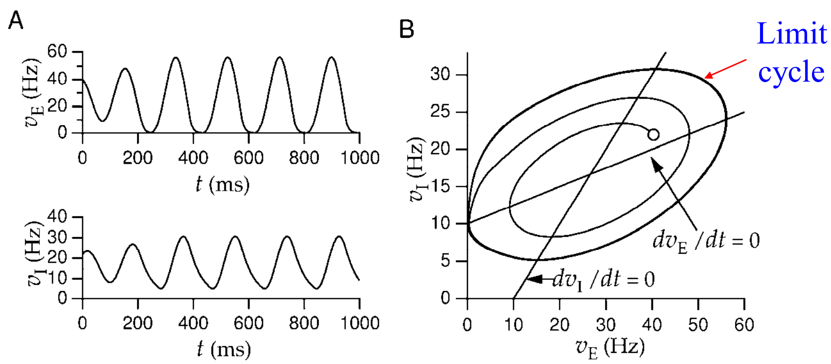
$$\tau_I\frac{dv_I}{dt} = -v_I + [0 \cdot v_I + v_E - 10]^+$$

Damped Oscillations in the Network



τ_I = 30 ms (negative real eigenvalue)

Unstable Behavior and Limit Cycle

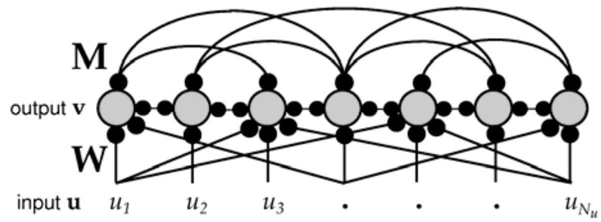


$\tau_I = 50$ ms (positive real eigenvalue)

So far, we have been analyzing networks with *fixed* sets of synaptic weights W and M

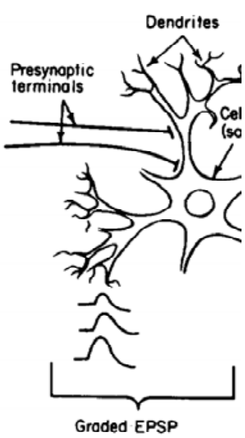
Can these be adapted in response to inputs?

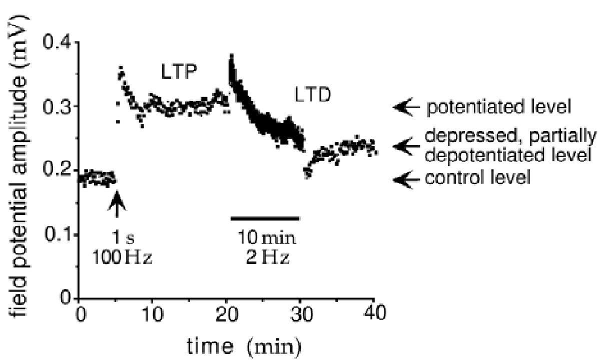
Plasticity and Learning: Adapting the Connections



- Question 1: How do we adapt the synaptic weights W and M to solve useful tasks?
- Question 2: How does the brain do it?

Synaptic Plasticity in the Brain

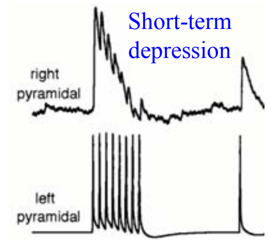




LTP = Long Term Potentiation
LTD = Long Term Depression

Other Forms of Plasticity in the Brain

- Short-Term Synaptic Plasticity
  - Short-term depression/facilitation
  - Dynamics may change on a long-term basis via LTP/LTD
- Changes to intrinsic excitability of cell
  - Density and distribution of various channels (ionic conductances)
  - Not well-studied
- Growth and morphological changes in dendrites
  - Not well-studied
- Addition of new neurons?
  - Hot topic of research these days…



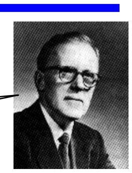
The Theory: Classification of Learning Algorithms

- **Unsupervised Learning**
  - Synapses adapted based solely on inputs
  - Network self-organizes in response to *statistical patterns* in input
  - Similar to Probability Density Estimation in statistics
- **Supervised Learning**
  - Synapses adapted based on inputs and desired outputs
  - External "teacher" provides desired output for each input
  - Goal: Function approximation
- **Reinforcement Learning**
  - Synapses adapted based on inputs and (delayed) reward/punishment
  - Goal: Pick outputs that *maximize total expected future reward*
  - Similar to optimization based on Markov decision processes

Let's start with Unsupervised Learning

Consider a single neuron receiving feedforward inputs from other neurons (e.g. from the retina)

The Grand-Daddy of Unsupervised Learning



- Rule hypothesized by Donald Hebb in 1949
- Hebb's learning rule:

"If neuron A frequently contributes to the firing of neuron B, then the synapse from A to B should be strengthened"

- Related Mantra: *Neurons that fire together wire together*
- Hebb's goal: Produce clusters of neurons ("*cell assemblies*") that fire together in response to a stimulus

Formalizing Hebb's Rule

- Consider a linear neuron: $v = \mathbf{w}^T \mathbf{u} = \mathbf{u}^T \mathbf{w}$
- Basic Hebb Rule: $\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{u}v$ (or $\mathbf{w} \leftarrow \mathbf{w} + \varepsilon \cdot \mathbf{u}v$)
- What is the average effect of this rule? $$\tau_w \frac{d\mathbf{w}}{dt} = \langle \mathbf{u}v \rangle_{\mathbf{u}} = Q\mathbf{w}$$
- Q is the input correlation matrix: $Q = \langle \mathbf{u}\mathbf{u}^T \rangle$

Variants of Hebb's Rule

- Pure Hebb only increases synaptic weights (LTP)
  - ➪ What about LTD?
- Covariance rules:

$$\tau_w \frac{d\mathbf{w}}{dt} = (\mathbf{u} - \boldsymbol{\theta}_{\mathrm{u}})v$$

(But: LTD also for no input and some output)

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{u}(v - \theta_v)$$

(But: LTD also for no output and some input)

Next Class: Unsupervised Learning

- Things to do:
  - Finish Chapter 8 and Start Chapter 10
  - Watch for the Last Homework (due May 24)
  - Start mini-project