Gameplan

- Plasticity and Learning
  - Types: Unsupervised, Supervised, and Reinforcement learning

- Today: Unsupervised Learning
  - Hebb rule and its variants (Covariance, Oja rule)
  - Mathematical formulation
  - Stability analysis of learning rules
So far, we have been analyzing networks with *fixed* sets of synaptic weights $W$ and $M$ (based on eigenvalues of $M$ etc.)

Can synaptic weights 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

How can we measure plasticity using electrodes?

A

Stim

Record

B

Increase in EPSP size for same input over time

Data from an Experiment

LTP = Long Term Potentiation
LTD = Long Term Depression

Hippocampus

(From: Citri and Malenka, Neuropsychopharmacology, 2008)
Long Term Potentiation (LTP)

LTP = Experimentally observed *increase* in synaptic strength that lasts for hours or days

![Diagram of LTP](image-source)

Long Term Depression (LTD)

LTD = Experimentally observed *decrease* in synaptic strength that lasts for hours or days

![Diagram of LTD](image-source)
Hebb’s Learning Rule

If neuron A repeatedly takes part in firing neuron B, then the synapse from A to B is strengthened.

“Neurons that fire together wire together!”

Image Source: Wikimedia Commons

Formalizing Hebb’s Rule

+ Consider a single linear neuron with steady state output:
  \[ v = w \cdot u = w^T u = u^T w \]

+ Basic Hebb Rule: \( \tau_w \frac{dw}{dt} = uv \)

Discrete Implementation:

\[ \tau_w \frac{w(t + \Delta t) - w(t)}{\Delta t} = uv \quad \text{ (or } w(t + \Delta t) = w(t) + \frac{\Delta t}{\tau_w} uv) \]

\[ w_{i+1} = w_i + \varepsilon \cdot u v \quad \text{ (or } \Delta w = \varepsilon \cdot u v) \]
What is the average effect of the Hebb rule?

- **Hebb Rule**: \( \tau_w \frac{dw}{dt} = uv \)

- Average effect of the rule:
  \[
  \tau_w \frac{dw}{dt} = \langle uv \rangle_u = \left\langle uu^T w \right\rangle_u = \left\langle uu^T \right\rangle_u w = Qw
  \]

- Q is the input correlation matrix: \( Q = \left\langle uu^T \right\rangle_u \)

---

Covariance Rule

- **Hebb rule only increases synaptic weights (LTP)**
  - What about LTD?

- **Covariance rule**: \( \tau_w \frac{dw}{dt} = u(v - \langle v \rangle) \) (Note: LTD for low or no output given some input)

- Average effect of the rule:
  \[
  \tau_w \frac{dw}{dt} = \langle u(v - \langle v \rangle) \rangle_u = \left\langle u(u^T - \langle u \rangle^T)w \right\rangle_u = \left( \left\langle uu^T \right\rangle - \langle u \rangle \langle u \rangle^T \right)w = Cw \quad (C \text{ is the input covariance matrix } \left\langle uu^T \right\rangle - \langle u \rangle \langle u \rangle^T)\]
Are these learning rules stable?

- **Does** \( w \) **converge to a stable value or explode?**
  - Look at what happens to the length of \( w \) over time

- **Hebb rule:** \( \tau_w \frac{dw}{dt} = uv \)
  
  \[
  \frac{d\|w\|^2}{dt} = 2w^T \frac{dw}{dt} = 2w^T (uv / \tau_w) = \frac{2}{\tau_w} v^2 > 0 \quad \text{w grows without bound!}
  \]

- **Covariance rule:** \( \tau_w \frac{dw}{dt} = u(v - \langle v \rangle) \)
  
  \[
  \frac{d\|w\|^2}{dt} = 2w^T \frac{dw}{dt} = 2w^T (u(v - \langle v \rangle) / \tau_w) = \frac{2}{\tau_w} (v^2 - \langle v \rangle^2)
  \]
  
  Averaging RHS, \( \frac{d\|w\|^2}{dt} = \frac{2}{\tau_w} (\langle v^2 \rangle - \langle v \rangle^2) = \frac{2}{\tau_w} \sigma_v^2 > 0 \quad \text{w grows without bound!} \]

---

Oja’s Rule for Hebbian Learning

- **Oja’s rule:** \( \tau_w \frac{dw}{dt} = uv - \alpha v^2 w \quad (\alpha > 0) \)

- **Stable?**
  
  \[
  \frac{d\|w\|^2}{dt} = 2w^T \frac{dw}{dt} = \frac{2}{\tau_w} w^T (uv - \alpha v^2 w) = \frac{2}{\tau_w} (v^2 - \alpha v^2 w^T w)
  \]
  
  i.e., \( \tau_w \frac{d\|w\|^2}{dt} = 2v^2 (1 - \alpha \|w\|^2) \)

  At steady state: \( \|w\|^2 = \frac{1}{\alpha} \quad \text{i.e.,} \quad \|w\| = \frac{1}{\sqrt{\alpha}} \)

  \( w \) does not grow without bound, i.e.,
  Oja’s rule is stable!
Summary: Hebbian Learning

- **Hebb rule:**
  \[ \tau_{\text{w}} \frac{dw}{dt} = uv \]
  Unstable (unless constraint on \( \|w\| \) is imposed)

- **Covariance rule:**
  \[ \tau_{\text{w}} \frac{dw}{dt} = u(v - \langle v \rangle) \]
  Unstable (unless constraint on \( \|w\| \) is imposed)

- **Oja’s rule:**
  \[ \tau_{\text{w}} \frac{dw}{dt} = uv - \alpha v^2 w \]
  Stable \( \|w\| \to \frac{1}{\sqrt{\alpha}} \)

What does Hebbian Learning do anyway?

- Start with the averaged Hebb rule: \( \tau_{\text{w}} \frac{dw}{dt} = Qw \)

- How do we solve this equation to find \( w(t) \)?
  
  : Eigenvectors to the rescue (again)!

- Write \( w(t) \) in terms of eigenvectors of \( Q \): \( w(t) = \sum_i c_i(t)e_i \)

- Substitute in Hebb rule diff. eq. and simplify as before:
  \[ \tau_{\text{w}} \frac{dc_i}{dt} = \lambda_i c_i \quad \text{i.e.,} \quad c_i(t) = c_i(0) \exp(\lambda_i t / \tau_{w}) \]

  \[ w(t) = \sum_i c_i(t)e_i = \sum_i c_i(0) \exp(\lambda_i t / \tau_{w})e_i \]

  For large \( t \), largest eigenvalue term dominates; \( w(t) \propto e_1 \)

  (For Oja’s rule: \( w(t) = \frac{e_1}{\sqrt{\alpha}} \))
The Brain can do Statistics!*  
Hebbian Learning implements *Principal Component Analysis* (PCA)

Hebb Rule  
Input mean = (0,0)  

Hebb Rule  
Input mean = (2,2)  

Covariance Rule  
Input mean = (2,2)

Hebbian learning learns a weight vector aligned with the principal eigenvector of input correlation/covariance matrix (i.e., direction of maximum variance)

*See a previous lecture for “The Brain can do Calculus!”

Next Class: Unsupervised to Supervised Learning

✦ Things to do:  
✦ Finish Chapter 8 and Start Chapter 10  
✦ Homework 3 due on Sunday, February 19  
✦ Start mini-project