Synaptic Plasticity

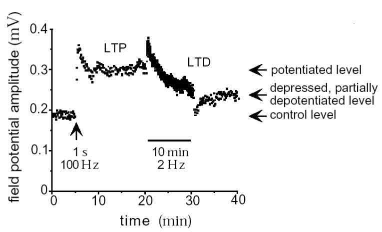
Dayan and Abbott (2001) Chapter 8

Instructor: Yoonsuck Choe; CPSC 644 Cortical Networks

Introduction

- Activity-dependent synaptic plasticity:
  - underlies learning and memory, and
  - plays a crucial role in neural circuit development.
- Donald Hebb: Hebb rule for synaptic plasticity (1949)
  - neuron A contributes to firing of neuron B, then
  - synapse between A and B should be strengthened.
  - Subsequent activation of A will lead to stronger activation of B.
- Hebb's rule only increases synaptic strength. It can be generalized to weaken strength if neuron A repeatedly fails to activate B.

Biophysics of Synaptic Plasticity



- Plasticity is found in many brain regions: hippocampus, cortex, cerebellum, etc.
- Plot above shows field potential recordings from CA1 region in rat hippocampus.
  - High-frequency stimulation leads to long-term potentiation (LTP).
  - Low-frequency stimulation leads to long-term depression (LTD).
  - Consistent with Hebb rule.
  - Postsynaptic concentration of Ca^{2+} ions play a role in LTP and LTD.

Functional Modes of Synaptic Plasticity

Types of learning:

- Unsupervised learning
- Supervised learning
- Reinforcement learning

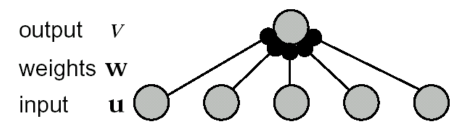
Types of synaptic plasticity:

- Hebbian synaptic plasticity
- Non-Hebbian synaptic plasticity: e.g., anti-Hebbian (decrease strength when co-activated).

Stability and Competition

- Increasing synaptic plasticity is a positive feedback process: Uncontrolled growth possible if unchecked.
- Dealing with unbounded growth:
  - Impose a saturation constraint: $0 \leq w \leq w_{\max}$: Possible problem of every weight turning $w_{\max}$.
  - Synaptic competition: Some weaken while some strengthen.

Network Model with Firing Rate Neurons



from Chapter 7

- Input vector $\mathbf{u}$
- Weight vector $\mathbf{w}$
- Output (postsynaptic activity) v

$$\tau_r \frac{dv}{dt} = -v + \mathbf{w} \cdot \mathbf{u} = -v + \sum_{b=1}^{N_u} w_b v_b,$$

or, after reaching steady state (set the above to 0):

$$v = \mathbf{w} \cdot \mathbf{u}$$

Basic Hebb Rule: Correlation-Based

- Simplest form has:

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u},$$

where τ_w is the time constant that controls the rate of change in w (learning rate).

- Ensemble averaging ($\langle \cdot \rangle$) over the inputs:

$$\tau_w \frac{d\mathbf{w}}{dt} = \langle v\mathbf{u} \rangle$$

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{Q} \cdot \mathbf{w} \text{ or } \tau_w \frac{dw_b}{dt} = \sum_{b'=1}^{N_u} Q_{bb'} w_{b'}, \text{ where}$$

$$\mathbf{Q} = \langle \mathbf{uu} \rangle \text{ or } Q_{bb'} = \langle u_b u_{b'} \rangle.$$

Unbounded Growth in Basic Hebb Rule

- Length of weight vector:

$$|\mathbf{w}|^2 = \mathbf{w} \cdot \mathbf{w} = \sum_b w_b^2.$$

Dot product of

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u}$$

and $\mathbf{w}$ gives:

$$\tau_w \frac{d|\mathbf{w}|^2}{dt} = 2v^2, \text{ given}$$

$$\frac{d|\mathbf{w}|^2}{dt} = 2\mathbf{w} \cdot \frac{d\mathbf{w}}{dt} \text{ and } \mathbf{w} \cdot \mathbf{u} = v.$$

- Note $v \geq 0$, so the above always increases (unless $v = 0$).

Discrete Updating Rule for Hebbian Learning

- Commonly used discrete update rule is:

$$\mathbf{w} \rightarrow \mathbf{w} + \epsilon \mathbf{Q} \cdot \mathbf{w},$$

where ϵ is analogous to $\frac{1}{\tau_w}$ in the continuous version.

- Even simpler implementation is:

$$\mathbf{w} \rightarrow \mathbf{w} + \epsilon v \mathbf{u},$$

i.e., no ensemble averaging.

Covariance Rule

- We want to allow a single rule to allow both increase and decrease in synaptic weight.

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

where θ_v is a threshold. Synaptic weight will decrease if $v < \theta_v$ and increase if $v > \theta_v$.

- An alternative is to put the threshold on the input side:

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

- If $\boldsymbol{\theta}_u = \langle \mathbf{u} \rangle$, we get

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{C} \cdot \mathbf{w}, \quad \text{where}$$

$$\mathbf{C} = \langle (\mathbf{u} - \langle \mathbf{u} \rangle)(\mathbf{u} - \langle \mathbf{u} \rangle) \rangle = \langle \mathbf{u}\mathbf{u} \rangle - \langle \mathbf{u} \rangle^2 = \langle (\mathbf{u} - \langle \mathbf{u} \rangle)\mathbf{u} \rangle.$$

Depression under Covariance Rule

- Homosynaptic depression: depression when nonzero input and $v < \theta_v$

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

- Heterosynaptic depression: depression when input is inactive and $v > 0$

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

- Implicit point: No input or output activity is required for LTD to happen.

Instability of the Covariance Rule

- The covariance rule is unstable despite the threshold:

$$\tau_w \frac{d|\mathbf{w}|^2}{dt} = 2v(v - \langle v \rangle),$$

where the time average of RHS is proportional to

$$\langle v^2 \rangle - \langle v \rangle^2,$$

which is positive (it's the variance of v).

BCM Rule

Bienenstock, Cooper, and Munro (1982).

- Synaptic plasticity requires both pre- and postsynaptic activity:

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

- Unstable like Hebb rule if θ_v is kept fixed.
- Condition for stability is:

$$\tau_\theta \frac{d\theta_v}{dt} = v^2 - \theta_v,$$

  where threshold adaptation rate τ_θ is typically smaller than τ_w.
- Sliding threshold implements synaptic competition: Increase in one synaptic weight will increase output v, thus it will increase threshold, making other synpases hard to adapt.

Preventing Unbounded Growth: Normalization

- Directly work on the weights rather than altering the threshold.
- Assumption is that increase in one synaptic weight should be balanced by the decrease in other synaptic weights.
- Thus, global constraints are needed:
  - Hold total sum of weights constant.
  - Constrain the sum of squares of the weights.

Synaptic Normalization: Subtractive

- Add a subtractive term in weight update:

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u} - \frac{v(\mathbf{n} \cdot \mathbf{u})\mathbf{n}}{N_u},$$

  where N_u is the length of $\mathbf{u}$, and $\mathbf{n} = (1, 1, 1, ...1)$, so $\sum w_b = \mathbf{n} \cdot \mathbf{w}$.
- This is a rigid constraint, since the sum of weights $\mathbf{n} \cdot \mathbf{w}$ does not change:

$$\tau_w \frac{d\mathbf{n} \cdot \mathbf{w}}{dt} = v\mathbf{n} \cdot \mathbf{u} \left(1 - \frac{\mathbf{n} \cdot \mathbf{n}}{N_u}\right) = 0.$$

- Biological basis is unclear.

Synaptic Normalization: Multiplicative

- Oja's rule (Oja, 1982)

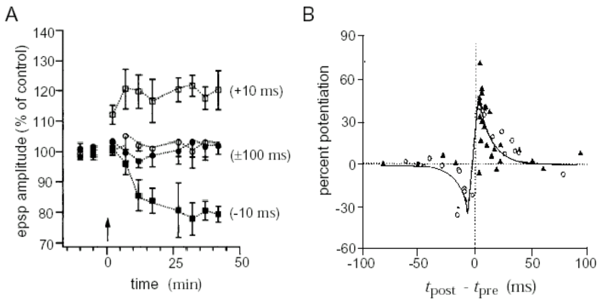
$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u} - \alpha v^2 \mathbf{w},$$

  with a positive constant α.
- It is based more on a theoretical argument than biological.
- Stability can be analyzed as before:

$$\tau_w \frac{d|\mathbf{w}|^2}{dt} = 2v^2(1 - \alpha|\mathbf{w}|^2).$$

  The steady state value of $|\mathbf{w}|^2$ becomes $1/\alpha$ (set the RHS to 0 and solve for $|\mathbf{w}|^2$).
- In other words, the length of the weight vector is held constant.

Timing-Based Rules



- Plasticity is time-dependent: Spike Timing Dep. Plast. (STDP)
- Presynaptic spike time t_{pre} and postsynaptic spike time t_{post}:
  - If post fires first then pre, $t_{\mathrm{post}} - t_{\mathrm{pre}} < 0$: pre did not cause post to fire.
  - If pre fires first then post, $t_{\mathrm{post}} - t_{\mathrm{pre}} > 0$: pre did cause post to fire.

Timing-Based Rule

- STDP applied to firing rate models:

$$\tau_w \frac{d\mathbf{w}}{dt} = \int_0^\infty d\tau (H(\tau)v(t)\mathbf{u}(t-\tau) + H(-\tau)v(t-\tau)\mathbf{u}(t)),$$

  where $H(\tau)$ takes a shape similar to the plot B in the previous page, depending on the sign of τ ($\mathrm{sign}(H(\tau)) = \mathrm{sign}(\tau)$).
- STDP is more naturally applied to spiking neuron models.

Unsupervised Learning

- Variants of Hebbian learning can be understood in the context of unsupervised learning.
- Major area of research: cortical map formation.
  - Orientation map, ocular dominance map, spatial frequencey map, etc. etc.
  - Activity-dependent (learning) and/or activity-independent (genetically determined)?

Hebbian Learning and Principal Eigenvector

- Diagonalize the correlation matrix $\mathbf{Q}$:

$$\mathbf{Q} \cdot \mathbf{e}_\mu = \lambda_\mu \mathbf{e}_\mu$$

  where $\mathbf{e}_\mu$ is an eigenvector (mutually orthogonal) and λ_μ is an eigenvalue ($\mu = 1, 2, ..., N_u$).
- For correlation and covariance matrices, all eigenvalues are real and nonnegative.
- We can express any N_u-dimensional vector as a linear combination of the N_u eigenvectors $\mathbf{e}_\mu$. So,

$$\mathbf{w}(t) = \sum_{\mu=1}^{N_u} c_\mu(t)\mathbf{e}_\mu,$$

  where c_μ are the coefficients.

Principal Eigenvector (cont'd)

- From

$$\mathbf{w}(t) = \sum_{\mu=1}^{N_u} c_\mu(t)\mathbf{e}_\mu,$$

$$\mathbf{w} = [c_1, c_2, \ldots c_\mu \ldots c_{N_u}] \begin{bmatrix} [& \mathbf{e}_1 &] \\ [& \mathbf{e}_2 &] \\ & \ldots & \\ [& \mathbf{e}_\mu &] \\ & \ldots & \\ [& \mathbf{e}_{N_u} &] \end{bmatrix} = \mathbf{cE}$$

we get

$$c_\mu(t) = \mathbf{w}(t) \cdot \mathbf{e}_\mu$$

since

$$\mathbf{wE}^{-1} = \mathbf{c}, \text{ and } \mathbf{E}^{-1} = \mathbf{E}^T.$$

Principal Eigenvector (cont'd)

- Plugging

$$\mathbf{w}(t) = \sum_{\mu=1}^{N_u} c_\mu(t)\mathbf{e}_\mu \quad \text{into} \quad \tau_w \frac{d\mathbf{w}}{dt} = \mathbf{Q}\cdot\mathbf{w} \quad \text{we get}$$

$$\tau_w \sum_{\mu=1}^{N_u} \frac{dc_\mu(t)}{dt}\mathbf{e}_\mu = \mathbf{Q}\cdot\sum_{\mu=1}^{N_u} c_\mu(t)\mathbf{e}_\mu.$$

Multiply both sides with $\mathbf{e}_\nu$, and $\mathbf{e}_\mu \cdot \mathbf{e}_\nu = \delta_{\mu\nu}$:

$$\tau_w \frac{dc_\mu}{dt} = c_\mu(t)(\mathbf{Q}\cdot\mathbf{e}_\mu)\cdot\mathbf{e}_\mu \quad \text{becomes} \quad \tau_w \frac{dc_\mu}{dt} = c_\mu(t)\lambda_\mu\mathbf{e}_\mu\cdot\mathbf{e}_\mu.$$

Since $\mathbf{Qe} = \lambda\mathbf{e}$

$$\tau_w \frac{dc_\mu}{dt} = c_\mu(t)\lambda_\mu, \quad \text{so, we get}$$

$$c_\mu(t) = c\exp\left(\frac{\lambda_\mu t}{\tau_w}\right), \quad \text{where } \mathbf{w}(0)\cdot\mathbf{e}_\mu = c_\mu(0), \quad \text{so } c = \mathbf{w}(0)\cdot\mathbf{e}_\mu.$$

$$\mathbf{w}(t) = \sum_{\mu=1}^{N_u} \exp\left(\frac{\lambda_\mu t}{\tau_w}\right)(\mathbf{w}(0)\cdot\mathbf{e}_\mu)\,\mathbf{e}_\mu.$$

Principal Eigenvector (cont'd)

$$\mathbf{w}(t) = \sum_{\mu=1}^{N_u} \exp\left(\frac{\lambda_\mu t}{\tau_w}\right)(\mathbf{w}(0)\cdot\mathbf{e}_\mu)\,\mathbf{e}_\mu.$$

- Thus, for large t, the vector term with the highest λ_μ factor ($\mu = 1$ if eigenvalues have been sorted) will dominate, so

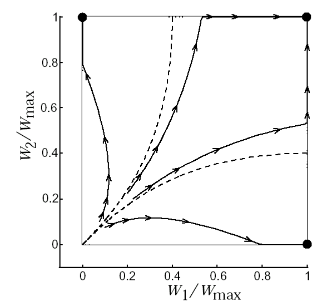
$$\mathbf{w} \propto \mathbf{e}_1.$$

- Finally, we get

$$v \propto \mathbf{e}_1 \cdot \mathbf{u}$$

which is the projection of the input vector along the principal eigenvector of the correlation/covariance matrix.

Principal Eigenvector: Issues



Principal eigenvector $\mathbf{e}_1 = (1, -1)/\sqrt{2}$

- The proportionality relation $v \propto \mathbf{e}_1 \cdot \mathbf{u}$ conceals the large exponential factor, which can grow without bound.
- Saturation constraint can help, but it can prevent the weight update to converge to the principal eigenvector (see figure above), depending on the initial condition.

Principal Eigenvector: Use of Oja's Rule

- Oja's rule (Oja, 1982) can be used to prevent unlimited growth:

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u} - \alpha v^2 \mathbf{w}.$$

- The rule gives $\mathbf{w} = \mathbf{e}_1/\sqrt{\alpha}$ as $t \to \infty$.

Principal Eigenvector: Use of Subtractive Normalization

- Averaging

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u} - \frac{v(\mathbf{n}\cdot\mathbf{u})\mathbf{n}}{N_u},$$

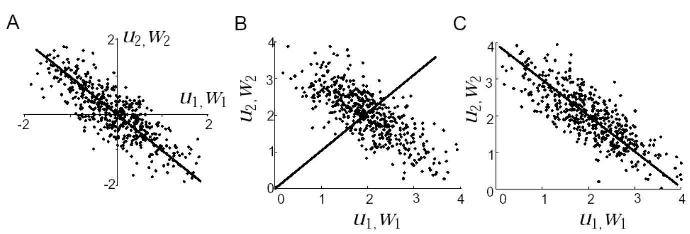
over input samples gives:

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{Q}\cdot\mathbf{w} - \frac{(\mathbf{w}\cdot\mathbf{Q}\cdot\mathbf{n})\mathbf{n}}{N_u}.$$

- Growth of $\mathbf{w}$ is unaffected by the second term if $\mathbf{e}_\mu \cdot \mathbf{n} = 0$. If $\mathbf{e}_\mu \cdot \mathbf{n} \neq 0$ weight will grow without bound.
- If principal eigenvector of $\mathbf{Q}$ is proportional to $\mathbf{n}$, $\mathbf{Q}\cdot\mathbf{e}_1 - (\mathbf{e}_1\cdot\mathbf{Q}\cdot\mathbf{n})\mathbf{n}/N = 0$, so principal eigenvector is unaffected by the learning rule. Also, $\mathbf{e}_\mu \cdot \mathbf{n} = 0$ for $\mu \geq 2$, so

$$\mathbf{w}(t) = (\mathbf{w}(0)\cdot\mathbf{e}_1)\mathbf{e}_1 + \sum_{\mu=2}^{N_u} \exp\left(\frac{\lambda_\mu t}{\tau_w}\right)(\mathbf{w}(0)\cdot\mathbf{e}_\mu)\mathbf{e}_\mu.$$

Hebbian Learning for PCA



- A: correlation rule, zero mean
- B: correlation rule, non-zero mean
- C: covariance rule, non-zero mean.