Learning the average

Sebastian Seung

Synaptic plasticity

- Potentiation: increase in strength
- Depression: decrease in strength
- Long-term: > 1 hour
- Activity-dependent: activity leaves an "impression" on synapses.
- More studied at excitatory synapses

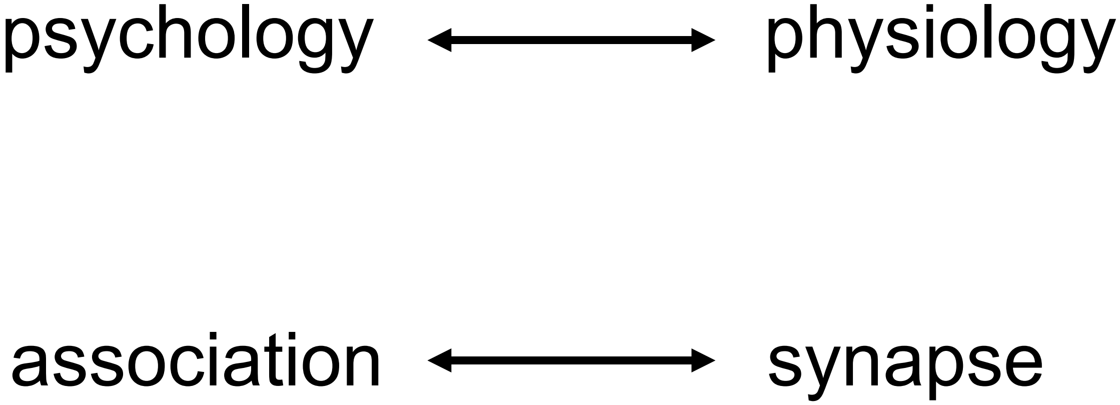
Hebbian synaptic plasticity

- At some synapses, coincident spiking of the presynaptic and postsynaptic neurons causes long-term potentiation.
- Such synapses are said to be Hebbian.

Donald Hebb (1949)

- When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A’s efficiency, as one of the cells firing B, is increased.

Associative memory

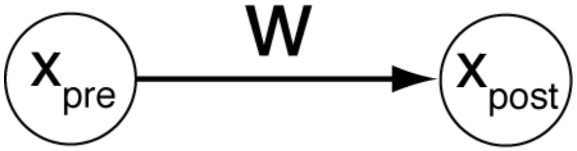


psychology ↔ physiology

association ↔ synapse

Hebb rule

- Multiplicative interaction between presynaptic and postsynaptic activity.



$$\Delta w = \eta x_{\text{pre}} x_{\text{post}}$$

Hypothesis: Hebbian synaptic plasticity enables a binary neuron to compute the mean of the stimuli that activate it.

Hebb rule

$$\Delta w_i = \eta y x_i$$

- vector form

$$\Delta \mathbf{w} = \eta y \mathbf{x}$$

- problem: this diverges

Hebb rule with weight decay

$$\Delta \mathbf{w} = \eta y (\mathbf{x} - \mathbf{w})$$

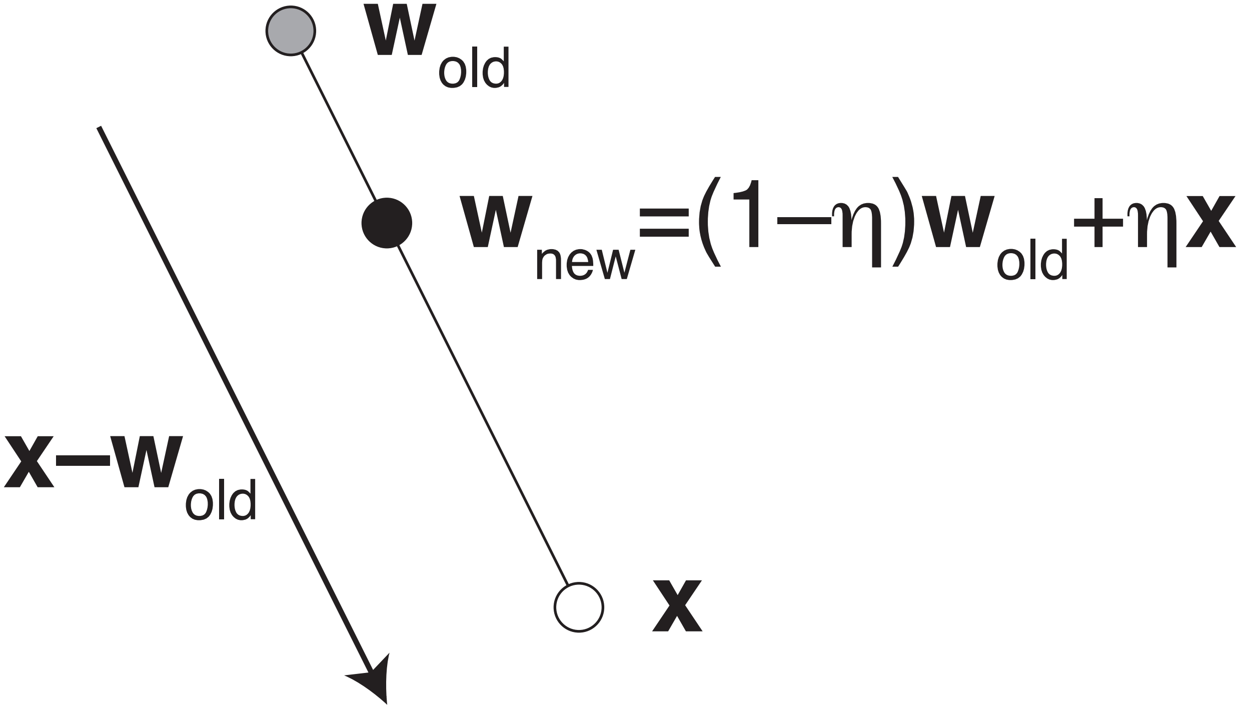
Case of constant activity

- Consider an LT neuron, and suppose it is always active $y = 1$

- The Hebb rule becomes $\Delta\mathbf{w} = \eta(\mathbf{x} - \mathbf{w})$

- The new weight vector is a linear interpolation between the old one and the input vector $\mathbf{w}' = (1 - \eta)\mathbf{w} + \eta\mathbf{x}$

Linear interpolation


$\mathbf{w}_{\text{old}}$
$\mathbf{w}_{\text{new}}=(1-\eta)\mathbf{w}_{\text{old}}+\eta\mathbf{x}$
$\mathbf{x}-\mathbf{w}_{\text{old}}$
$\mathbf{x}$

Average velocity approximation

$$\Delta\mathbf{w} \approx \langle \Delta\mathbf{w} \rangle$$

- The learning dynamics becomes

$$\Delta\mathbf{w} \approx \eta(\langle \mathbf{x} \rangle - w)$$

- The steady state is

$$\mathbf{w} = \langle \mathbf{x} \rangle$$

The Hebb rule with weight decay is a gradient-based optimization algorithm.

Cost function

$$e\left(\mathbf{w},\mathbf{x}\right)=\frac{1}{2}\left|\mathbf{w}-\mathbf{x}\right|^2$$

$$\frac{\partial}{\partial\mathbf{w}}e\left(\mathbf{w},\mathbf{x},y\right)=\mathbf{w}-\mathbf{x}$$

- Online update $\Delta\mathbf{w}=-\eta\frac{\partial e}{\partial\mathbf{w}}$

Gradient learning

- Batch update is gradient descent on

$$E(\mathbf{w}) = \sum_a e(\mathbf{w}, \mathbf{x}^a) = \frac{1}{2} \sum_a \left| \mathbf{w} - \mathbf{x}^a \right|^2$$

- Online update with randomized order is stochastic gradient descent.

Online vs. batch update

- Update w after each example.

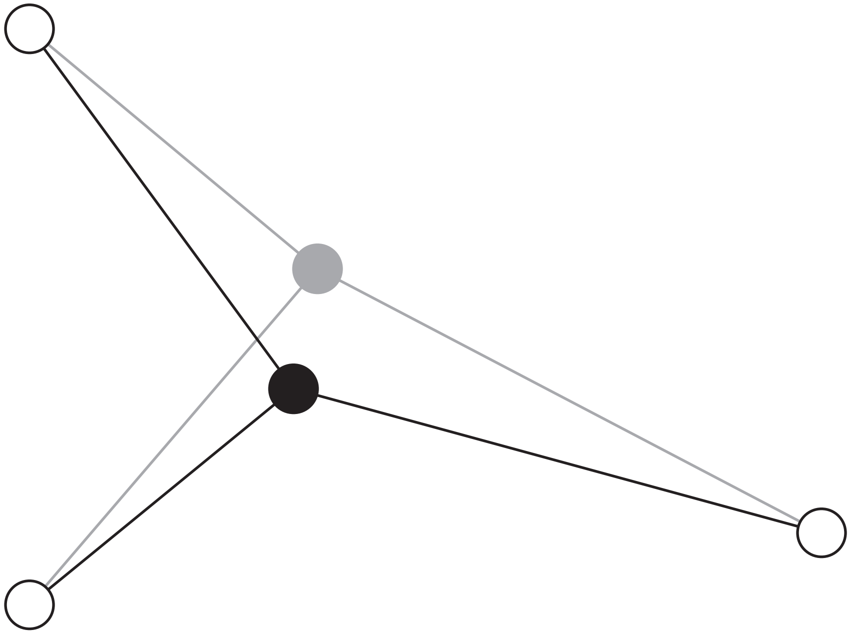
$$\Delta \mathbf{w} = \eta \left(\mathbf{x} - \mathbf{w} \right)$$

- Update w after the whole batch of examples.

$$\Delta \mathbf{w} = \eta \sum_{a} \left(\mathbf{x}^{a} - \mathbf{w} \right)$$

A rubber band computer

$$E(\mathbf{w}) = \frac{1}{m}\sum_{\mu=1}^{m}\frac{1}{2}|\mathbf{w}-\mathbf{x}_\mu|^2$$



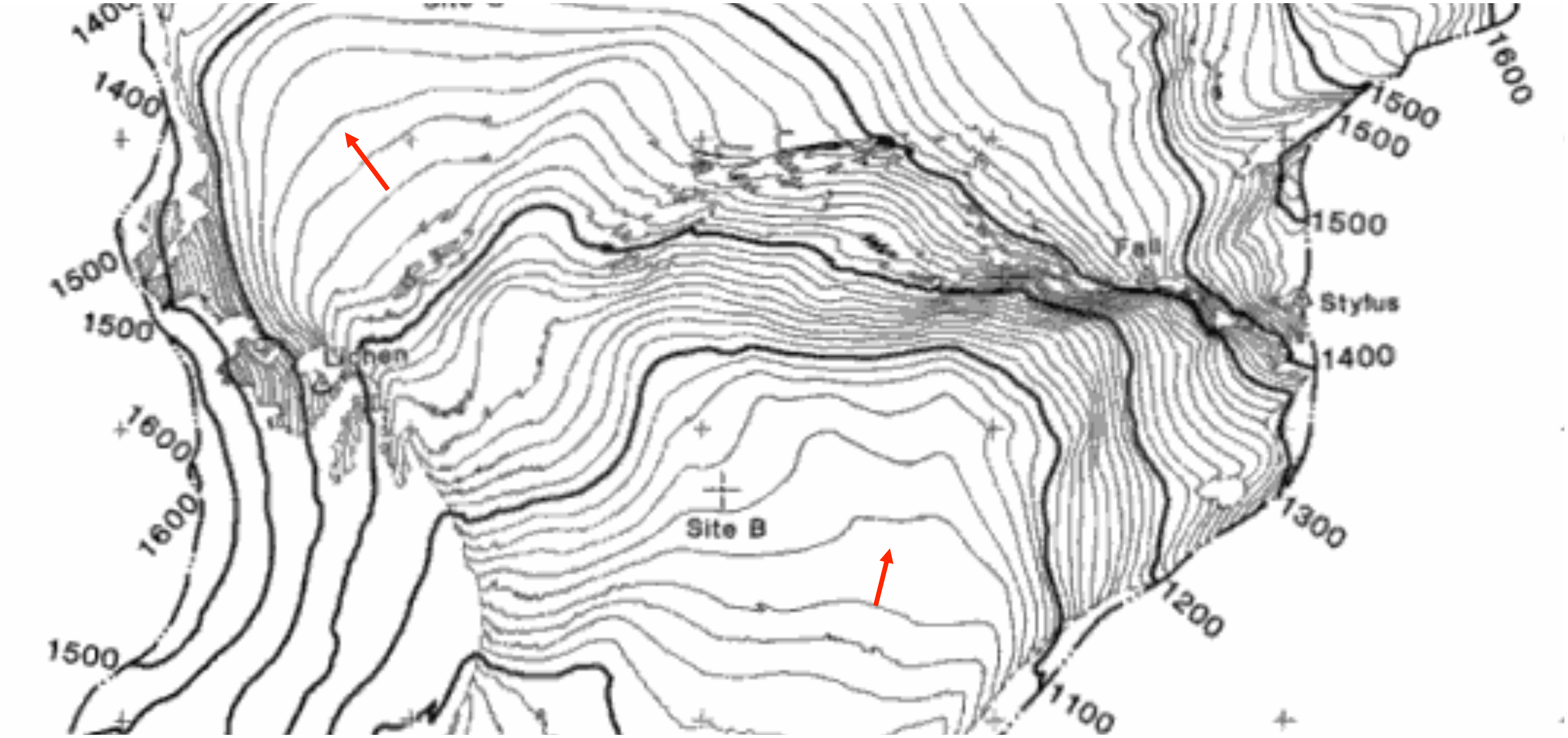
Gradient

- Vector of partial derivatives

$$\nabla E = \frac{\partial E}{\partial \mathbf{w}} = \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_N} \right)$$

- Direction of steepest ascent ∇E
- Direction of steepest descent $-\nabla E$

The gradient is perpendicular to the contour lines (level sets)



Wolverine Glacier, Alaska

Gradient descent

$$\frac{d\mathbf{w}}{dt} = -\frac{\partial E}{\partial \mathbf{w}}$$

- Converges to a local minimum of the cost function E
  - provable under weak assumptions
- Simple optimization algorithm

Gradient descent

- Discrete time version

$$\Delta\mathbf{w} = -\eta \frac{\partial E}{\partial \mathbf{w}}$$

- Finite step size controlled by η
- Converges to a local minimum if η is small.

The sample mean maximizes likelihood

- Gaussian distribution

$$P_\mu(x) \propto \exp\left(-\frac{1}{2}|x-\mu|^2\right)$$

- Maximize

$$P_\mu(x_1)P_\mu(x_2)\cdots P_\mu(x_m)$$