Unsupervised and Representation Learning

Davide Bacciu

Dipartimento di Informatica
Università di Pisa
bacciu@di.unipi.it

Applied Brain Science - Computational Neuroscience (CNS)

Part I

Module Information

Objectives

Train computational neuroscience and machine learning specialists capable of

- understanding application challenges and choosing the right neural-network-based solutions
- designing computational models from biological memory mechanisms
- developing advanced applications using ML solutions

Expected Outcome

Students completing the module are expected to

- Gain in-depth knowledge of advanced hierarchical neural network architectures
- Learn state of the art unsupervised and representation learning algorithms
- Understand their theory and applications
- Be able to individually read, understand and discuss research works in the field

The course is targeted at

- Students specializing in
  - Machine learning and computational intelligence
  - Data mining, data sciences and information retrieval
  - Robotics, bionics, bioengineering
- Students seeking machine learning theses

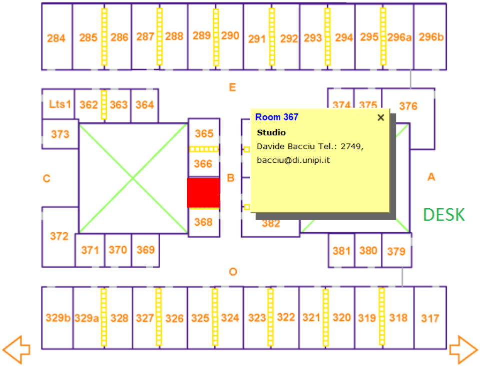
Topics

- Synaptic plasticity, memory and learning
  - Associative learning, competitive learning and inhibition
- Associative memory models
  - Hopfield networks
  - Boltzmann Machines
  - Adaptive Resonance Theory
- Representation learning and hierarchical models
  - Biological inspiration: sparse coding, pooling and information processing in the visual cortex
  - HMAX, CNN, Deep Learning

Instructor

Davide Bacciu

Email - `bacciu@di.unipi.it`
Tel - 050 2212749
Office - Room 367, Dipartimento di Informatica
Office hours - Monday 17-19 (email me!)



Schedule

Lectures

1. Unsupervised and representation learning (3h)
2. Associative Memories I - Hopfield networks (2h)
3. Associative Memories II - Boltzmann Machines (2h)
4. Adaptive Resonance Theory (1h)
5. Representation learning and hierarchical models (2h)
6. Deep Learning (3h)

Laboratory activity

1. Hands-on Lab I (3h)
2. Hands-on Lab II (2h)

Try to fit an additional hour to deep learning lecture for some extra lab

Homepage

Reference Webpage on Didawiki:

`http://didawiki.di.unipi.it/doku.php/bionics-engineering/computational-neuroscience/start`

Here you can find

- Course information
- Lecture slides
- Articles and course materials

You can subscribe to get RSS feeds on page updates

Reference Books

A classical reference book for Computational Neuroscience courses:

P. Dayan and L.F. Abbott, *Theoretical Neuroscience*, The MIT press (2001)

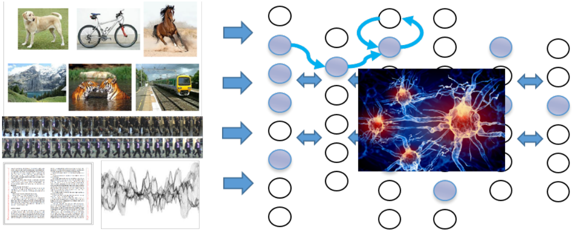
An alternative book covering similar topics and freely available online:

W. Gerstner, W.M. Kistler, R. Naud and L. Paninski, *Neuronal Dynamics: From single neurons to networks and models of cognition*, Cambridge University Press (2014)

Part II

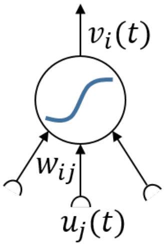
Unsupervised and Representation Learning

The Big Picture



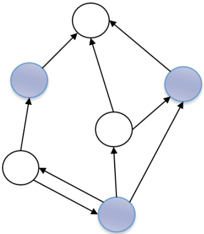
- Learning to encode complex/noisy input information in the activations of a neural network (representational learning)
- Requires a mechanism to reconfigure the synaptic response (plasticity)
- A computational approach through bio-inspiration

Reference Model



- A simple computational neuron abstraction
- Synaptic inputs u_j are integrated to determine activation v_i
- Synaptic weights w_{ij} and activation function F

- Neural network representing connected assemblies of computational neurons
- Distributed representation of stimuli

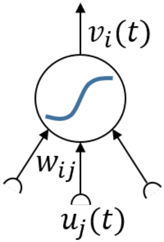


Learning in the Brain

Introducing relatively permanent changes in neuron behavior as a result of experience with stimuli

- Many different learning flavours
  - Perceptual
  - Stimulus-response
  - Motor
  - Relational
- Many adaptation mechanisms
  - Habituation
  - Sensitization
  - Priming
  - Conditioning
- A common underlying aspect ⇒ synaptic plasticity

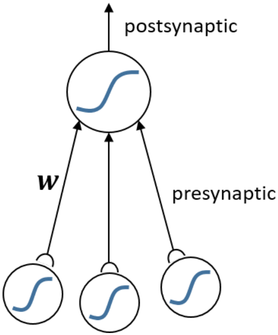
Synaptic Plasticity



- Synapses are characterized by their weight w_{ij}
- Determines the response of a postsynaptic neuron i to an action potential from presynaptic neuron j
- Assumed fixed so far
- Electrophysiological experiments show that the response (amplitude) is not fixed but can change over time
- Changes of the synaptic strength are called synaptic plasticity

Models of Synaptic Plasticity

- Synaptic plasticity depends on a variety of factors
  - Different causes: co-activation, repetition,...
  - Different effects: enhancement, depression
  - Different timescales: short-term, long-term,...
- Hebbian - Plasticity depends on both presynaptic and postsynaptic activity
- How do we define synaptic activity?
  - Firing rates
  - Spikes (action potentials)



Learning Paradigms

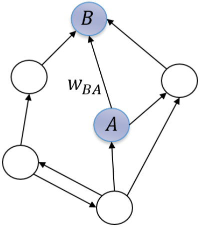
How synaptic plasticity is used as part of a training process to change the neural response?

- Unsupervised learning
  - Network responds to a series of inputs during training solely on the basis of intrinsic connections and dynamics
  - principal component analysis, density estimation, representation learning
- Supervised learning
  - A desired set of input-output relationship is imposed on the network by a teacher
  - Regression, classification, imitation learning
- Reinforcement learning
  - Network response is adjusted through a reward/punishment signal assessing performance on the task

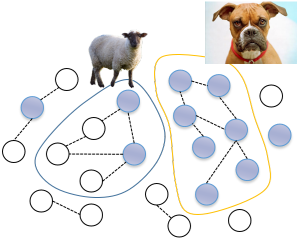
Hebbian Learning

The Organization of Behavior, 1949 (Donald Hebb)

When an axon of cell A is near enough to excite 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.

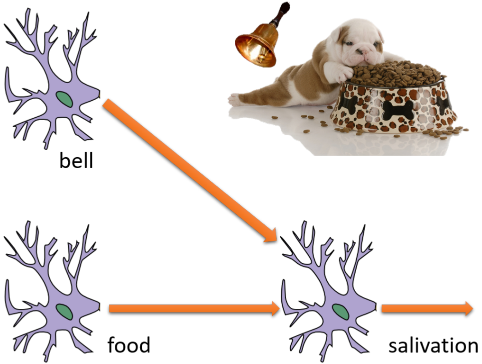


Neurons that fire together, wire together



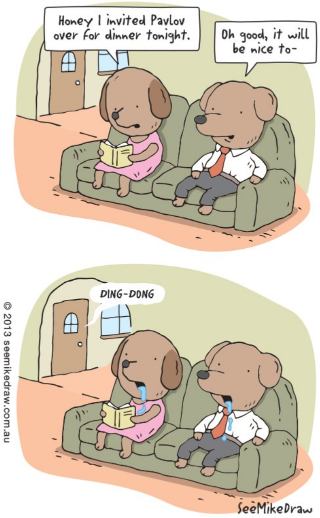
Self-organization of neuron assemblies

Hebb, Pavlov and his Dog



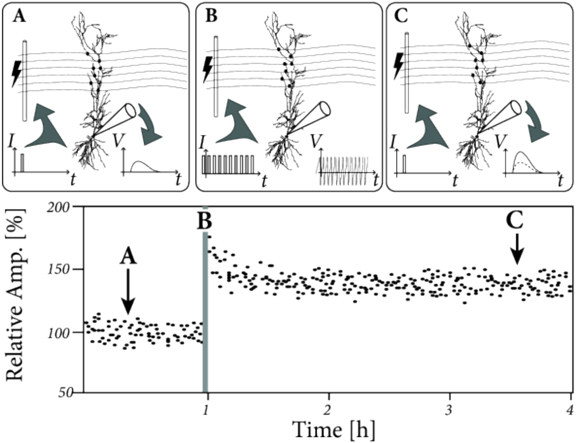
What happens now everytime a bell is rung?

Hebb, Pavlov and his Dog

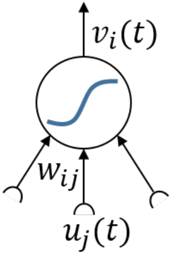


Long Term Potentiation (LTP)

A enhancement of the synaptic response whose effect are long-lasting (e.g. at least 10 minutes)



Firing Rate Neuron (Refresher)



- Models the steady-state output firing rate as

$$v_\infty = F(\mathbf{w} \cdot \mathbf{u})$$

- With a time-dependent input current, the firing-rate dynamics is

$$\tau_r \frac{dv}{dt} = -v + F(\mathbf{w} \cdot \mathbf{u})$$

- Refer to the simple linear model

$$\tau_r \frac{dv}{dt} = -v + \sum_{j=1}^{N_u} w_j u_j$$

at steady-state $v = \mathbf{w} \cdot \mathbf{u}$.

Relax the constraint of positive firing rates

Hebb Rule

- The basic Hebb rule

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

  where τ_w is the learning rate.

- In general, an averaged version would be preferred

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

  where $\langle \cdot \rangle$ averages on input patterns

- Inserting the linear firing-rate yields to the correlation rule

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{Q}\mathbf{w} \text{ s.t. } \mathbf{Q} = \langle \mathbf{u}\mathbf{u} \rangle$$

Hebb Rule and Postsynaptic Depression

- Basic Hebbian learning does not account for Long Term Depression (LTD)
- Synapse strength depresses if presynaptic activity is paired with low postsynaptic activation

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

$$\tau_w \frac{d\mathbf{w}}{dt} = v(\mathbf{u} - \theta_u) \quad \text{presynaptic}$$

where θ_v, θ_u switch LTD to LTP, e.g. $\theta_v = \langle v \rangle$

- When averaged both equivalent to the covariance rule

$$\tau_w \frac{d\mathbf{w}}{dt} = \mathbf{C}\mathbf{w} \text{ s.t. } \mathbf{C} = \langle (\mathbf{u} - \langle \mathbf{u} \rangle)(\mathbf{u} - \langle \mathbf{u} \rangle) \rangle$$

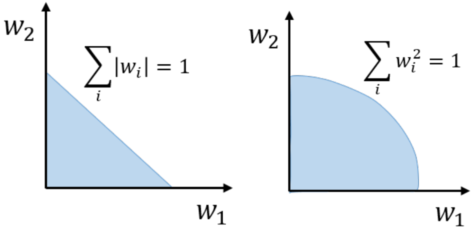
Basic Hebbian Learning is Unstable

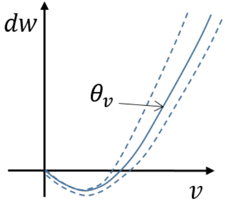
Positive feedback loop where activity and weights mutually reinforce

- Learning is unstable
  - Uncontrolled weight growth
  - Solution: Weight saturation constraints
- Synapses are updated independently
  - Poor selectivity to different inputs
  - Solution: Synaptic competition

Stabilization Strategies

(1) Synaptic normalization





(2) Nonlinear learning rules

Both strategies prevent unbounded weight growth as well as introduce synaptic competition

Synaptic normalization

Key Idea

- Add terms to the Hebb rules that depend explicitly on weights
- Assume a neuron can support only a fixed total amount of synaptic weights

Question: What machine learning practice does this recall?

- Normalization constraint can be imposed
  - Rigidly: at every time step
  - Dynamically: satisfied only asymptotically at the end of training
- Different normalization constraint and strategies can lead to (consistently) different training outcomes

Oja Rule (a.k.a. Multiplicative Normalization)

Sum-of-squares normalization term

$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u} - \alpha v^2 \mathbf{w} \;\; \text{s.t.} \;\; \alpha \geq 0$$

Stability is given since the length of the weight vector $\|\mathbf{w}\|_2^2$ will relax over time to $1/\alpha$, since

$$\tau_w \frac{d\|\mathbf{w}\|_2^2}{dt} = 2v^2(1 - \alpha\|\mathbf{w}\|_2^2)$$

Note that it also introduces synaptic competition (Why?)

Oja rule is local, dynamic but not biologically plausible

The BMC Rule

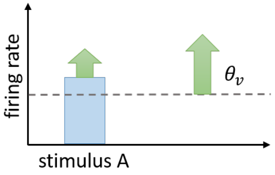
Bienenstock, Cooper and Munro (1982)

A more biologically plausible synaptic update

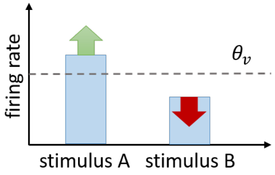
$$\tau_w \frac{d\mathbf{w}}{dt} = v\mathbf{u}(v - \theta_v)$$

A non-linear learning rule introducing a sliding threshold

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



No unbounded weight growth



Competition among stimuli

Unsupervised Learning

- A computational view of the effects of Hebbian synaptic plasticity in artificial neural networks
- Focus on training without a teacher signal
  - Learning to encode stimuli
  - Cortical maps
- Assess the effect of
  - Synaptic competition
  - Neuron competition
  - Inhibition

What does Oja Rule Learn?

When considering a linear neuron the Oja rule is

$$\frac{d\mathbf{w}}{dt} = \nu(\mathbf{u}v - v^2\mathbf{w}) = \nu(\mathbf{u}\mathbf{u}^T\mathbf{w} - \mathbf{w}^T\mathbf{u}\mathbf{u}^T\mathbf{w}\mathbf{w})$$

The expected value of $d\mathbf{w}$ (averaged on inputs) is

$$\frac{d\langle\mathbf{w}\rangle}{dt} = \nu\langle(\mathbf{u}\mathbf{u}^T\mathbf{w} - \mathbf{w}^T\mathbf{u}\mathbf{u}^T\mathbf{w}\mathbf{w})\rangle = \nu(\mathbf{Q}\mathbf{w} - (\mathbf{w}^T\mathbf{Q}\mathbf{w})\mathbf{w})$$

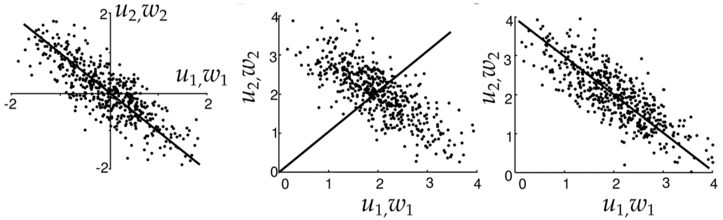
At convergence

$$\frac{d\langle\mathbf{w}\rangle}{dt} = 0 = \nu(\mathbf{Q}\mathbf{w} - (\mathbf{w}^T\mathbf{Q}\mathbf{w})\mathbf{w})$$
$$\Leftrightarrow \mathbf{Q}\mathbf{w} = (\mathbf{w}^T\mathbf{Q}\mathbf{w})\mathbf{w} = \lambda\mathbf{w}$$

The eigenvalue problem $\Rightarrow$ Eigenvectors of **Q** are potential solutions

Hebb, Oja and the PCA

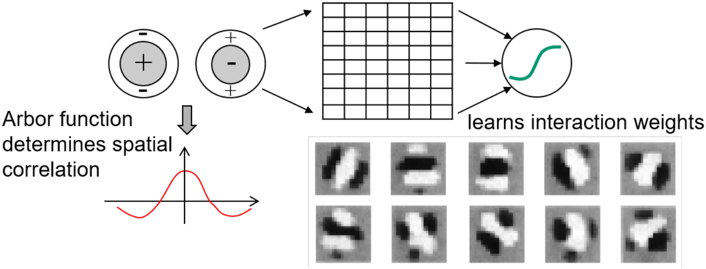
The fixed point of Oja rule (but also Hebb) is the largest eigenvector of **Q**



- Hebbian learning rotates weight vector to align with the principal eigenvector of input correlation/covariance matrix
- The weigh vector in Hebb rule has unbounded norm, while Oja rule learns a normalized weight vector

Competitive Learning
Synaptic Competition

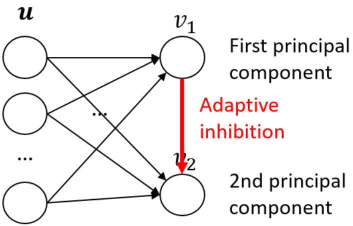
- Synaptic competition favours selectivity
- Hebbian rule with constraints preventing unconstrained growth explains orientation selectivity in primary visual cortex



Competitive Learning

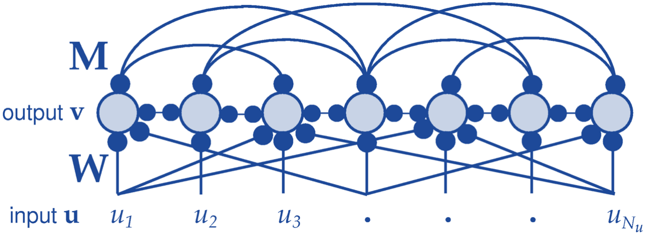
Principal Component Analysis

- Hebbian learning orient the weight vector so that the neuron is responsive to information in the principal component of data covariance/correlation
- Can we identify other principal components? How?



- Introduce competition between neurons
- Different neurons encode different stimuli
- Selectivity and diversity

Competitive Learning with Multiple Neurons



- Recurrent connections serve neuron differentiation
- Output activity

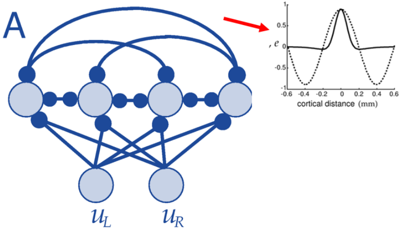
$$\tau_v \frac{d\mathbf{v}}{dt} = -\mathbf{v} + \mathbf{W}\mathbf{u} + \mathbf{M}\mathbf{v}$$

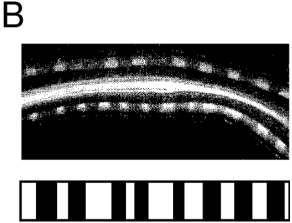
- Stable fixed point with steady-state output ($\rho(\mathbf{M}) < 1$)

$$\mathbf{v} = \mathbf{W}\mathbf{u} + \mathbf{M}\mathbf{v}, \mathbf{v} = \mathbf{K}\mathbf{W}\mathbf{u} \ \text{ and } \ \mathbf{K} = (\mathbf{I} - \mathbf{M})^{-1}$$

Ocular Dominance

Fix the recurrent connections by a periodic function (e.g. cos) on the position of the neuron in the cortex





Plastic feedforward, fixed recurrent connections

Competitive Hebbian Learning

A two step-process

1. Long-range competition (feedforward)

$$z_i = \frac{\left(\sum_j W_{ij} u_j\right)^\delta}{\sum_k \left(\sum_j W_{kj} u_j\right)^\delta}$$

2. Short-range cooperation between neighbors (recurrent)

$$v_i = \sum_{k \in Ne(i)} M_{ik} z_k$$

Purely linear units produce little differentiation among neurons (added nonlinearity δ)

Competitive Feature-based Hebbian Models

- Presynaptic inputs encode different features of the stimuli
- Neuron activation measures how closely the input matches its preferred stimulus
- Competitive activation phase

$$z_i = \frac{\exp\left(-\sum_j (u_j - W_{ij})^2/(2\sigma_j^2)\right)}{\sum_k \exp\left(-\sum_j (u_j - W_{kj})^2/(2\sigma_j^2)\right)}$$

- Cooperative mechanism to ensure nearby neurons have same selectivity
  - Self-organizing maps: $v_i = \sum_{k \in Ne(i)} M_{ik} z_k$
  - Elastic-net: $v_i = z_i$

Modeling Causality in Learning

Let's review Hebb's hypothesis

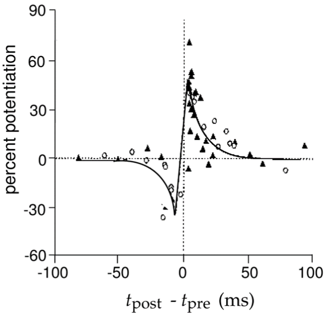
When an axon of cell A is near enough to excite cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic change takes place in one or both cells such that A's efficiency, as one of the cells firing B, is increased.

The interpretation of taking part is the key

- Relative timing between pre-synaptic and post-synaptic activity plays a critical role
- Proved experimentally and is also at the root of Hebb's original interpretation
- Spike-timing dependence in synaptic plasticity (STDP)

Approximate firing-rate model (in place of spiking neuron model)

STDP Reprise

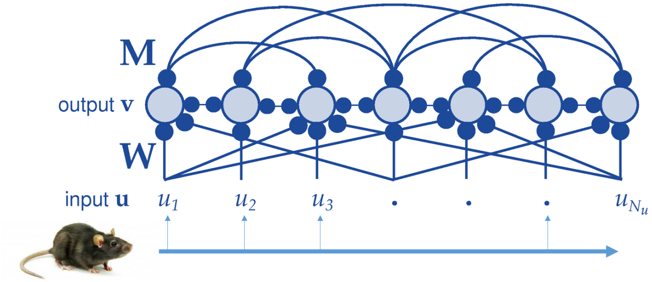


- LTP when post-synaptic occurs within 50msec from pre-synaptic spike
- LTD when pre-synaptic spike occurs within 50msec from post-synaptic action potential

$H(\tau) \Rightarrow$ Rate of synaptic modification when post-synaptic activity is separated from pre-synaptic by τmsec

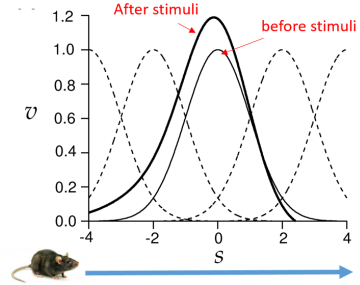
$$\tau_w \frac{d\mathbf{w}}{dt} = \int_0^\infty \big(\underbrace{H(\tau)v(t)\mathbf{u}(t-\tau)}_{\text{LTP}} + \underbrace{H(-\tau)v(t-\tau)\mathbf{u}(t)}_{\text{LTD}}\big)d\tau$$

Time-Dependent Plasticity with Multiple Neurons (I)



Keep **W** fixed and adapt recurrent connections **M** using time-dependent plasticity

Time-Dependent Plasticity with Multiple Neurons (II)



- Strengthening synapses between neurons that fire in close sequence (LTP)
- Central neuron tuning curve shifts denoting anticipatory response to pattern

(Much) More on recurrent networks for time-dependent tasks on the third module

Take Home Messages

- Synaptic plasticity is the mechanism underlying all learning scheme
- Hebbian learning
  - Promotes synapses between co-activated neurons (LTP)
  - Depresses synapses responsible for asynchronous activations (LTD)
  - Leads to principal component analysis, self-organizing maps, visual filters, ...
- Competition is essential to ensure
  - Selectivity and stability at synaptic level
  - Diversity between neurons
- Hebbian time-dependent plasticity allows learning sequential patterns

Things We Haven't Seen

- Anti-Hebbian Learning
  - Reducing synaptic strength as result of co-activation
- Timing-based plasticity and the spiking model
- Supervised Hebbian learning
  - Perceptron
  - Delta-rule

Next Lecture

- Associative memories
  - Red hammers and priming
  - Learning and recalling associations between stimuli/concepts
- Hopfield networks
  - An associative memory
  - A recurrent neural network
  - An energy-based model