Unsupervised learning

Overview

- Hebbian postulate
- Introducing competition through normalisation
- Topographic map formation and ocular dominance.

Hebb's postulate

- Supervised learning requires a teacher to provide error signals; how far can the nervous system get on its own?
- "When an axon of cell A is near enough to excite 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" (Hebb, 1949).
- AKA "cells that fire together wire together" or "out of sync lose the link".
- Detect correlations [over some small time window, say 50ms] between firing of cells.

$$\Delta w_j = \epsilon y x_j \qquad \text{where } y = f(\mathbf{w} \cdot \mathbf{x})$$

Correlation and covariance matrices

Often we average over inputs $\mathbf{x}$, assuming inputs change quicker than synaptic weights $\mathbf{w}$.
Correlation matrix $\mathbf{Q}_{ij} = \langle x_i x_j \rangle$ or $\mathbf{Q} = \langle \mathbf{x}\mathbf{x}^T \rangle$.

Covariance matrix $\mathbf{C}_{ij} = \langle (x_i - \mu_i)(x_j - \mu_j) \rangle$

Properties:

1. Real, symmetric matrix $\Rightarrow$ N orthogonal real eigenvectors.
2. Positive semi-definite: for any input $\mathbf{u}$, $\mathbf{u}^T \mathbf{C} \mathbf{u} \geq 0$.
3. All eigenvalues of a positive semidefinite matrix are non-negative.

Hebbian rule

Since we assume inputs change quicker than synaptic weights **w**:

$$
\begin{aligned}
\tau_w \frac{\mathrm{d}w_j}{\mathrm{d}t} &= \langle y x_j \rangle, \qquad y = \sum_i w_i x_i \\
&= \langle \sum_i w_i x_i x_j \rangle \\
&= \sum_i w_i \langle x_i x_j \rangle = \sum_i \mathbf{C}_{ji} w_i \\
\tau_w \frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} &= \mathbf{C}\mathbf{w}
\end{aligned}
$$

Variants on Hebbian rule

$$\text{activation rule} \qquad y = \mathbf{w} \cdot \mathbf{x}$$

$$\text{Hebb rule} \qquad \tau \frac{\mathrm{d}w_j}{\mathrm{d}t} = y x_j$$

$$\text{equivalently, for discrete update} \qquad \Delta w_j = \epsilon y x_j$$

This rule is unstable; for positive inputs and weights we get only growth of connections (examine $\frac{\mathrm{d}|\mathbf{w}|^2}{\mathrm{d}t}$). Can introduce decay of connections by thresholds either on input or output:

$$\text{postsynaptic threshold} \qquad \tau \frac{\mathrm{d}w_j}{\mathrm{d}t} = (y - \theta_y) x_j \qquad \theta_y = \langle y \rangle$$

$$\text{presynaptic threshold} \qquad \tau \frac{\mathrm{d}w_j}{\mathrm{d}t} = y (x_j - \theta_{x_j}) \qquad \theta_{x_j} = \langle x_j \rangle$$

However, these rules are still unstable.

Normalisation

Hebbian-based learning rules alone are unstable. Need some way to keep weights within bounds and introduce **competition**. Approaches:

1. Enforce limits on individual weights, e.g. [0,1].
2. Renormalise weights periodically to **rigidly** satisfy some constraint ($\sum_j w_j = K$ or $\sum_j w_j^2 = K$).

$$\tau_w \frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} = y\mathbf{x} - [\frac{y(\mathbf{n}\cdot\mathbf{x})}{\mathbf{n}\cdot\mathbf{n}}]\mathbf{n} \qquad \mathbf{n} = \text{vector of 1s}$$

subtractive normalisation: sub. k off each weight. $\frac{\mathrm{d}\sum_i w_i}{\mathrm{d}t} = \frac{\mathrm{d}(\mathbf{n}\cdot\mathbf{w})}{\mathrm{d}t} = 0.$

$$\tau_w \frac{\mathrm{d}\mathbf{w}}{\mathrm{d}t} = y\mathbf{x} - [\frac{y(\mathbf{n}\cdot\mathbf{x})}{\mathbf{n}\cdot\mathbf{w}}]\mathbf{w}$$

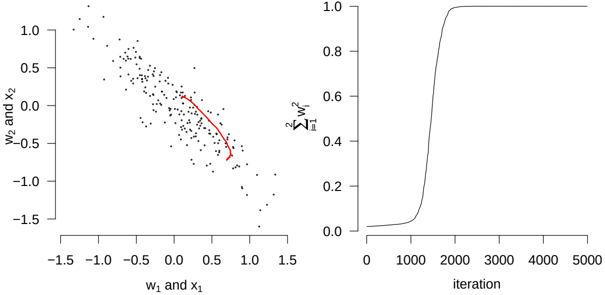
divisive normalisation: divide every weight by k.

Subtractive and divisive normalisation have different geometrical effects.

Oja rule as principal component analysis extractor

Add terms to learning rule so that constraints are **dynamically** enforced:

$$\tau \frac{d\mathbf{w}}{dt} = y\mathbf{x} - \alpha y^2 \mathbf{w}$$



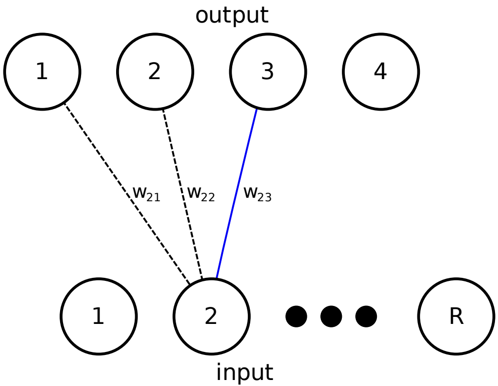
At s.s. $\mathbf{w}$ is maximal eigenvector of $\mathbf{Q}$.

This is Principal components analysis (PCA). Rule finds vector such that projection onto that vector maximises the variance of the responses. (nth PCA = nth largest eigenvector of correlation matrix of inputs.)

Extracting multiple principal components

What happens when we wire up multiple output neurons and use the Oja rule? Sanger rule (1989):

$$\Delta w_{ij} = \epsilon y_j (x_i - \sum_{k=1}^{j} y_k w_{ik})$$



Non-local update rule but reliable extraction of PCs in order.

Orientation selective receptive fields (Hancock et al. 1991)