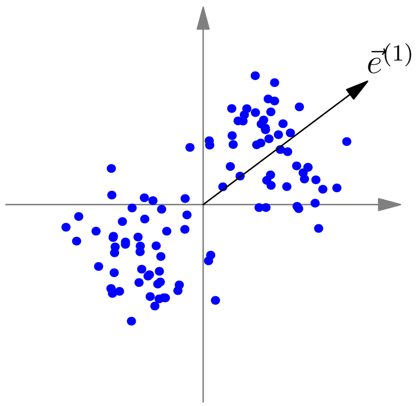
Principal Component Analysis

Given high-dimensional data, can we find a low-dimensional subspace still containing most of the information?

Idea behind PCA

Preserve only directions which have large variance!



Optimal dimensionality reduction using principal components

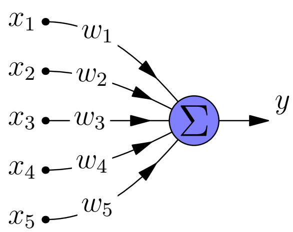
Karhunen Loève transform

- Find the subspace where the variance is largest
- Drop the rest

PCA — Principal Component Analysis
Technique:
Find the eigenvectors corresponding to the largest eigenvalues of the correlation matrix

Can a neural network perform PCA?

Self organizing linear unit



$$y = \sum_i x_i w_i$$

$$y = \vec{x}^T \vec{w} = \vec{w}^T \vec{x}$$

Hebbs rule:

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

Average change $\langle \Delta \vec{w} \rangle$

$$\langle \Delta \vec{w} \rangle = \eta \left\langle \vec{x}\vec{x}^T \vec{w} \right\rangle = \eta \mathcal{R} \vec{w}$$

$$\mathcal{R} = \left\langle \vec{x}\vec{x}^T \right\rangle = \text{Correlation Matrix}$$

What happens when using Hebbs rule repeatedly?

$$\langle \Delta \vec{w} \rangle = \eta \mathcal{R} \vec{w}$$

$\mathcal{R}$ has orthogonal eigenvectors: $\vec{e}^{(1)}, \vec{e}^{(2)}, \ldots$
Express $\vec{w}$ in components:

$$\vec{w} = \vec{w}^{(1)} + \vec{w}^{(2)} + \cdots = (\vec{w}^T \vec{e}^{(1)})\vec{e}^{(1)} + (\vec{w}^T \vec{e}^{(2)})\vec{e}^{(2)} + \cdots$$

$$\mathcal{R}\vec{w} = \mathcal{R}\vec{w}^{(1)} + \mathcal{R}\vec{w}^{(2)} + \cdots = \lambda_1 \vec{w}^{(1)} + \lambda_2 \vec{w}^{(2)} + \cdots$$

$\vec{w}$ increases without limit in the direction corresponding to the largest eigenvalue!

Oja's modified Hebb-rule

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

Converges to

$$|\vec{w}| = 1$$

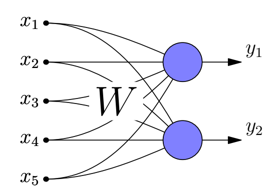
$\vec{w}$ will still be directed according to the largest eigenvalue

$$\langle \Delta \vec{w} \rangle = \eta \left(\left\langle \vec{x}\vec{x}^T \right\rangle \vec{w} - \left\langle \vec{w}^T \vec{x}\vec{x}^T \vec{w} \right\rangle \vec{w} \right) = \eta \left(\mathcal{R}\vec{w} - \underbrace{(\vec{w}^T \mathcal{R} \vec{w})}_{\lambda} \vec{w} \right)$$

Stability: $\langle \Delta \vec{w} \rangle = \vec{0}$

$$\mathcal{R}\vec{w} = \lambda \vec{w} \qquad \text{Eigenvector}$$

Oja's generalized method



$$\Delta W = \eta \vec{y} \left(\vec{x}^T - \vec{y}^T W \right)$$

GHA — Generalized Hebbian Algorithm

Converges to the subspace spanned by the eigenvectors of the largest eigenvalues

Solves PCA
Automatically performs an optimal dimensionality reduction

Trick to make the principal components ordered

GHA according to Oja:

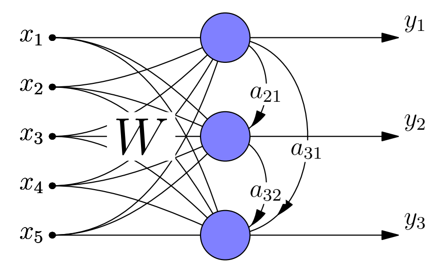
$$\Delta w_{ji} = \eta y_j \left(x_i - \sum_{k=1}^{N} y_k w_{ki} \right)$$

GHA according to Sanger:

$$\Delta w_{ji} = \eta y_j \left(x_i - \sum_{k=1}^{j} y_k w_{ki} \right)$$

Orders the principal components according to the size of the eigenvalues

APEX method — Adaptive Principal Component Extraction

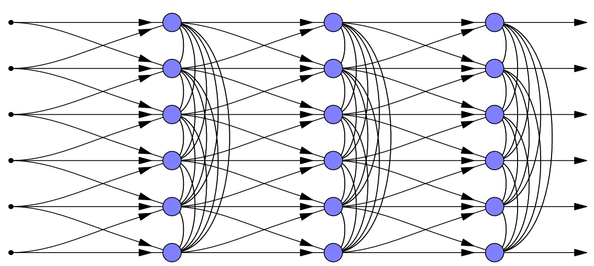


$$\Delta w_{ji} = \eta y_j \left(x_i - y_j w_{ji} \right) \qquad \Delta a_{ji} = \eta y_j \left(-y_i - y_j a_{ji} \right)$$

w_{ji} — Hebbian synapses $\qquad$ a_{ji} — Anti-Hebbian synapses

Similar to Lateral Inhibition

Linskers model of the visual system



Spontaneously evolves center-surround, orientation sensors, m.m.