Adaptive Intelligence

Prof. Eleni Vasilaki

The BCM rule

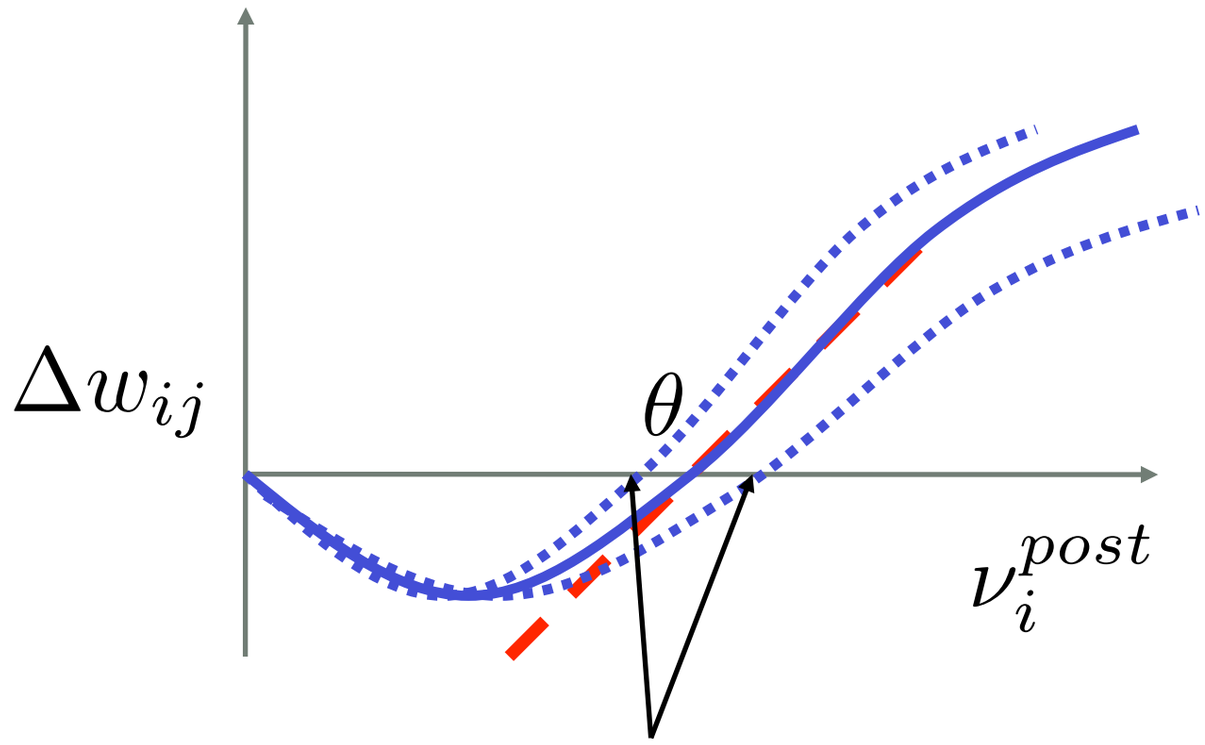
$$\Delta w_{ij} = \alpha \Phi \left(\nu_i^{post} - \theta \right) \nu_j^{pre}$$

Bienenstock, Cooper, Munro 1982

$$\Delta w_{ij} = \alpha \left(\nu_i^{post} - \theta \right) \nu_j^{pre}$$

presynaptically gated

$$\theta = f(< \nu_i^{post} >)$$

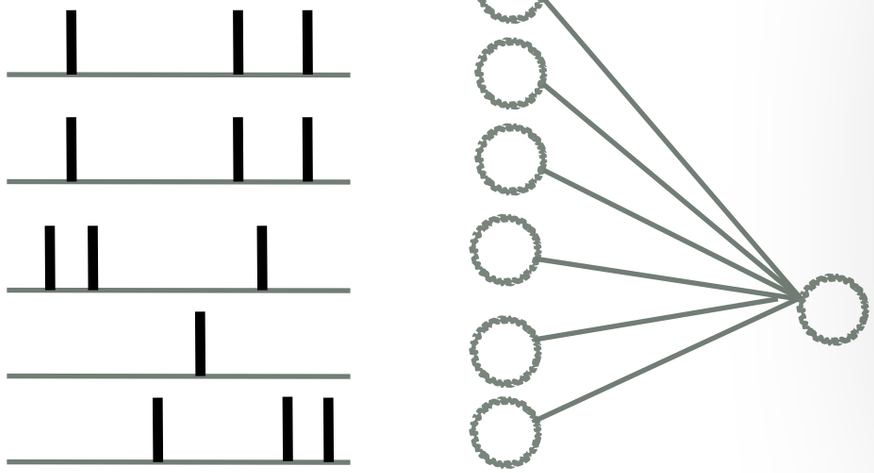


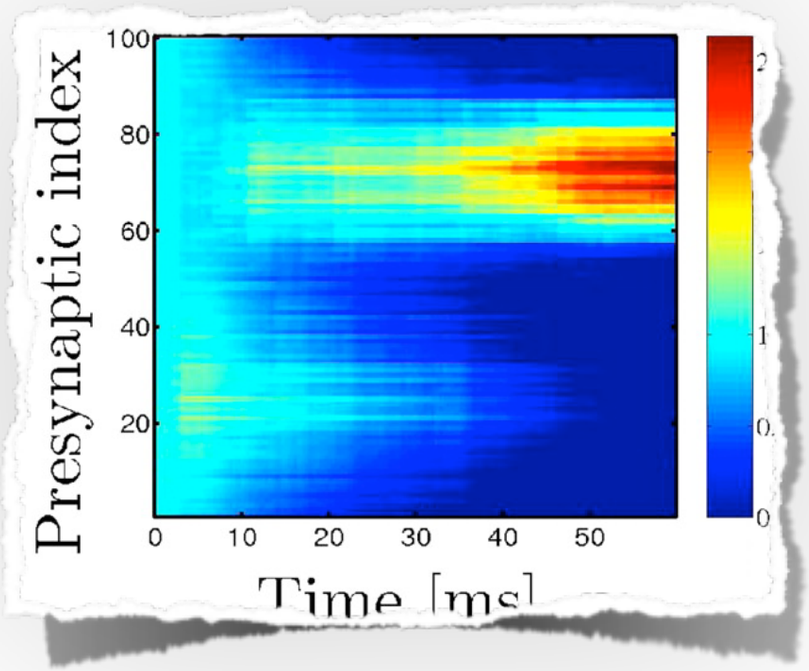
homeostasis

BCM

Jointly variant (correlated) input

Uncorrelated input





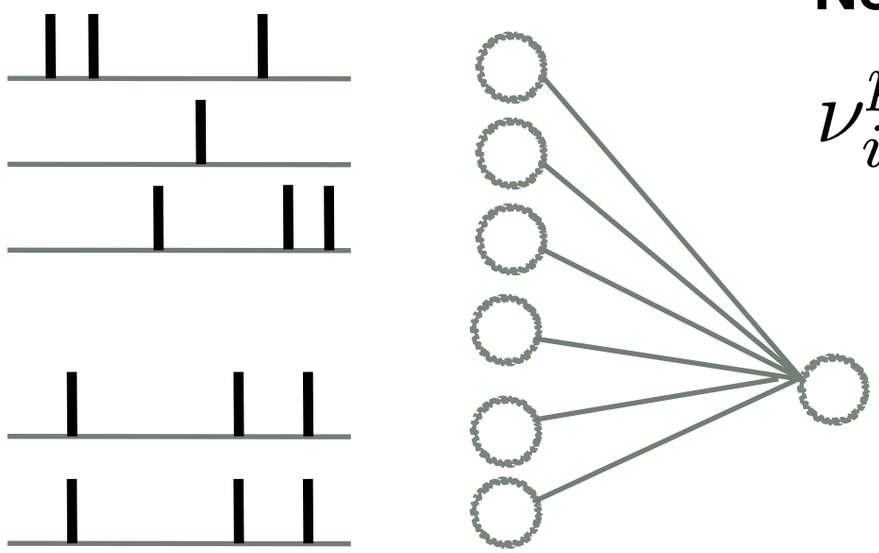
Weight evolution

$$\Delta w_{ij} = \alpha \Phi \left(\nu_i^{post} - \theta \right) \nu_j^{pre}$$

Is this a hebbian rule?

Detects where the ‘interesting’ input occurs (correlations): some synapses strengthened at the expense of others

BCM Rule



Neuron model

$$\nu_i^{post} = \sum_k w_{ik} \nu_k^{pre}$$

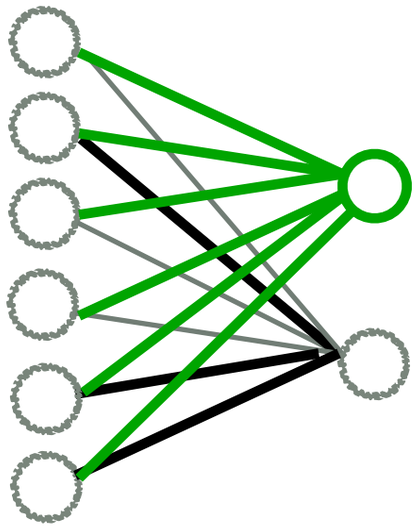
$$\Delta w_{ij} = \alpha \Phi \left(\nu_i^{post} - \theta \right) \nu_j^{pre}$$

Assume two (presynaptic) groups of 10 neurons each. All weights = 1 and threshold *θ= 20Hz .*

1. Group 1 fires at 3 Hz, then group 2 at 1 Hz. What happens to the weights?
2. Group 1 fires at 3 Hz, then group 2 at 2.5 Hz. What happens?
3. As in 2, but make theta a function of the averaged rate. What happens?

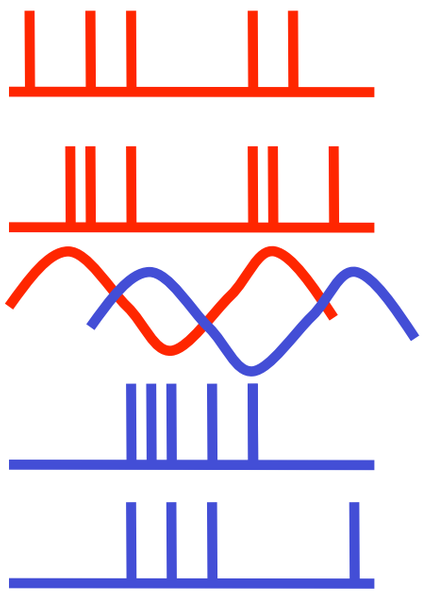
BCM leads to specialised neurons

Initial: random connections



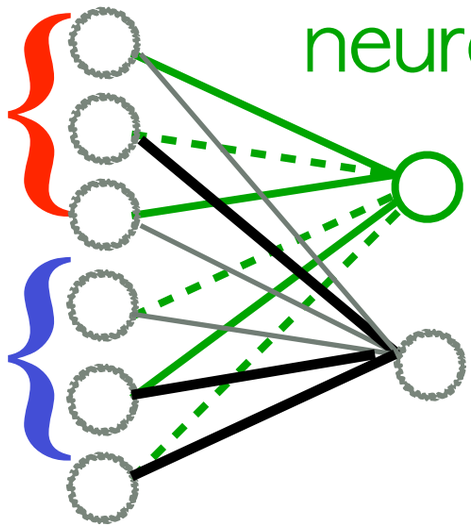
unselective neurons

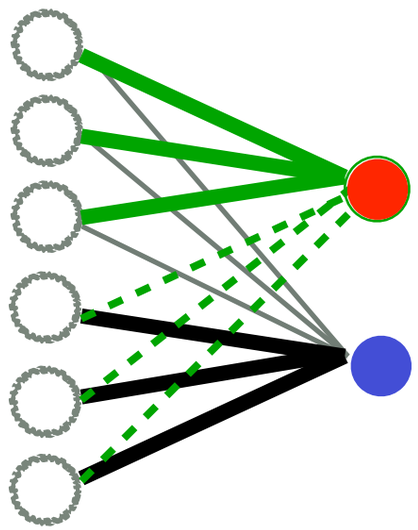
Receptive field development



Correlated input

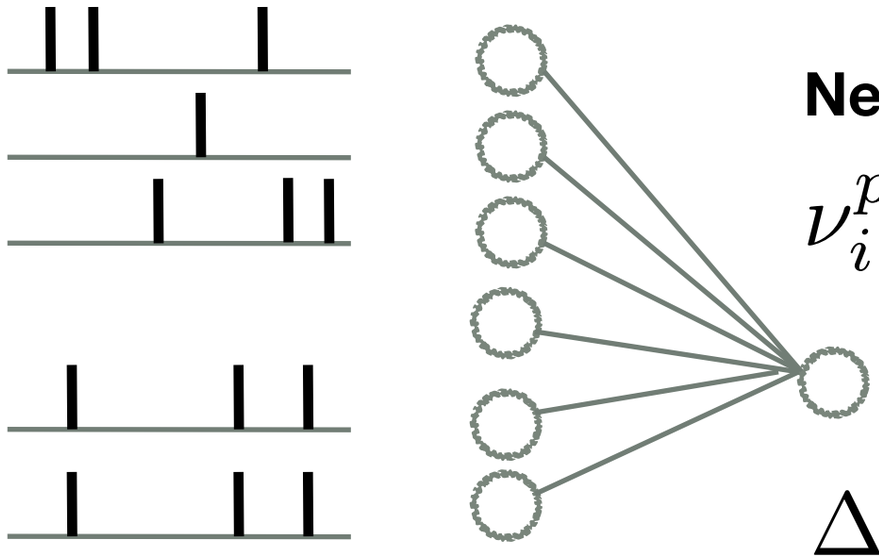
output neurons





output neurons specialize

Oja's rule



Neuron model

$$\nu_i^{post} = \sum_k w_{ik} \nu_k^{pre}$$

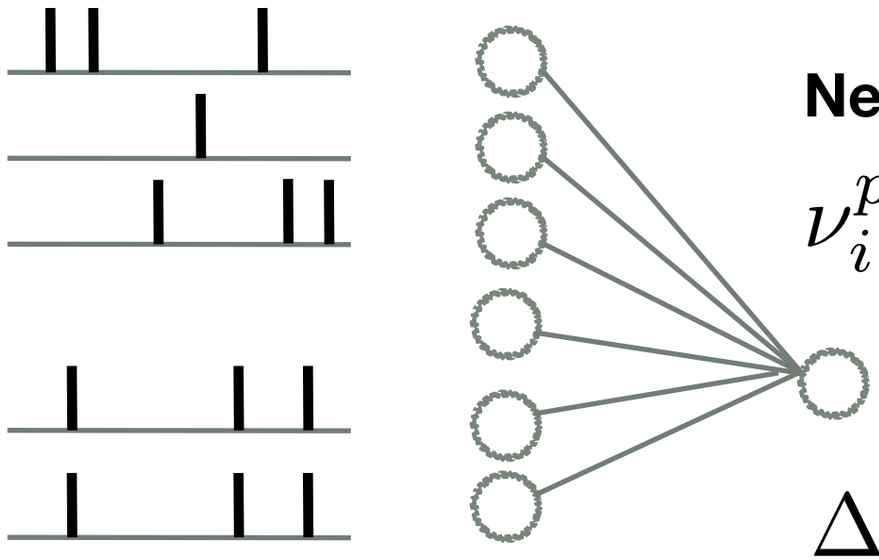
$$\Delta w_{ij} = \alpha \; \nu_i^{post} \nu_j^{pre} - \alpha \; w_{ij} \left(\nu_i^{post} \right)^2$$

Is this a hebbian rule?

What do we know from the earlier results on the "minimal" Hebb rule ?

What advantages this rule have over the "minimal" Hebb rule ?

Oja's rule



Neuron model

$$\nu_i^{post} = \sum_k w_{ik} \nu_k^{pre}$$

$$\Delta w_{ij} = \alpha \; \nu_i^{post} \nu_j^{pre} - \alpha \; w_{ij} \left(\nu_i^{post} \right)^2$$

Show that:

$$\langle \Delta w_{ij} \rangle = \alpha \left(\sum_k w_{ik} \left\langle \nu_k^{pre} \nu_j^{pre} \right\rangle - w_{ij} \sum_{k,n} w_{ik} \left\langle \nu_k^{pre} \nu_n^{pre} \right\rangle w_{in} \right)$$

Theory of Oja's rule

- Oja's rule coverages to a weight vector with the following properties:

    1. length equal to 1.

    2. is an eigenvector of the correlation matrix.

❖ Convergence means weight changes equal 0.

Oja’s rule

- Oja’s rule coverages to a weight vector with the following properties:

  1. length equal to 1.

❖ Convergence means weight changes equal 0.

$$\frac{d\mathbf{w}}{dt} = \alpha\left(y\mathbf{x} - y^2\mathbf{w}\right)$$

Oja's rule

$$\frac{d\mathbf{w}}{dt} = \alpha\left(y\mathbf{x} - y^2\mathbf{w}\right)$$

$$\mathbf{w}^T\frac{d\mathbf{w}}{dt} = \mathbf{w}^T\alpha\left(y\mathbf{x} - y^2\mathbf{w}\right)$$

$$\frac{1}{2}\frac{d|\mathbf{w}|^2}{dt} = \alpha y^2\left(1 - |\mathbf{w}|^2\right)$$

Oja's rule

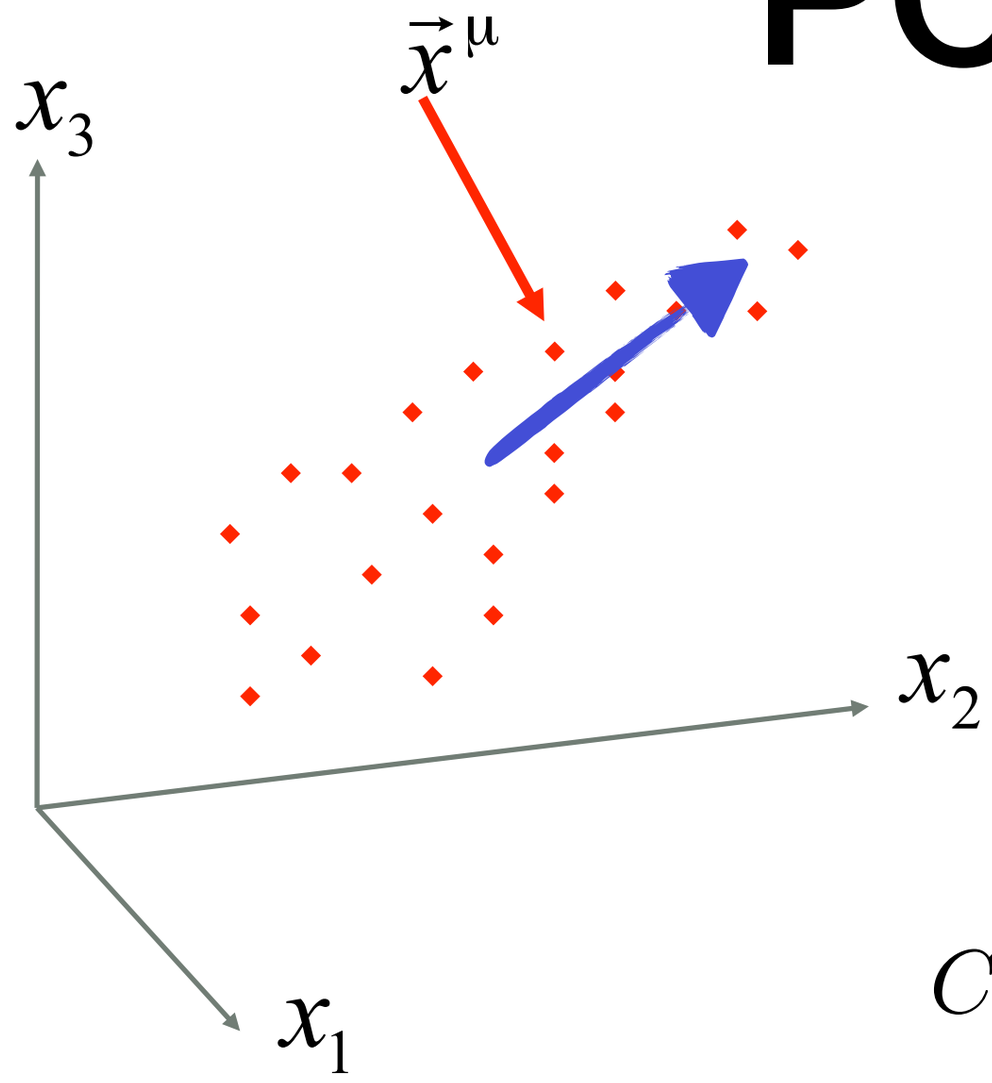
$$\frac{1}{2}\frac{d|\mathbf{w}|^2}{dt} = \alpha y^2 \left(1 - |\mathbf{w}|^2\right)$$

$$0 = \alpha y^2 \left(1 - |\tilde{\mathbf{w}}|^2\right)$$

$$|\tilde{\mathbf{w}}|^2 = 1$$

$$\frac{1}{2}\frac{d|\mathbf{w}|^2}{dt} = \alpha y^2 \left(1 - (\,|\tilde{\mathbf{w}}|^2 + \epsilon\,)\right) = -\alpha y^2 \epsilon$$

PCA



Correlation:

$$C_{kj} = \langle x_k x_j \rangle = \frac{1}{P} \sum_{\mu=1}^{P} x_k^{\mu} x_j^{\mu}$$

Covariance:

$$C_{kj}^{0} = \langle (x_k - \langle x_k \rangle)(x_j - \langle x_j \rangle) \rangle$$

$$\mathbf{C^0 e_g{}^n} = \lambda^{\mathbf{n}} \mathbf{e_g{}^n} \qquad \lambda^1 > \lambda^2 > \ldots > \lambda^N$$

$\mathbf{e_g{}^1}$ **Principal Component**

PCA

1. Calculate signal means, subtract means from signals.

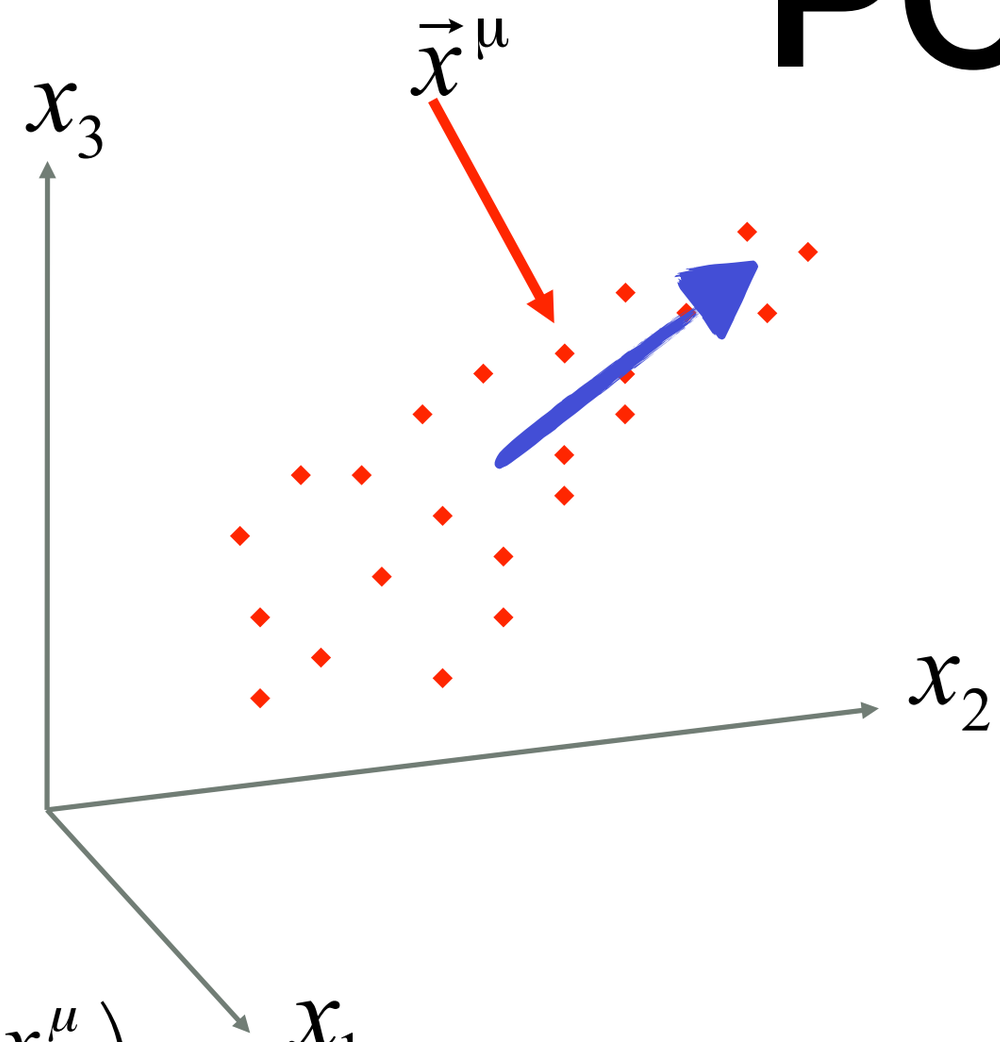
2. Write down the covariance formula. Calculate the covariance matrix.

3. Find the eigenvalues and eigenvectors. Make sure that the length of the eigenvectors is 1!

4. Decide how many eigenvectors you keep. From the eigenvectors form the “Feature Vector”.

5. Derive the new data:
   FinalData=RowFeatureVectorxRowDataAdjust.
   RowFeatureVector has the eigenvectors in the rows with the eigenvector with the highest eigenvalue On TOP.
   RowDataAdjust contains the zero mean data in each column, with each row holding a separate dimension.

PCA



PCA Tutorial:
Lindsay I Smith

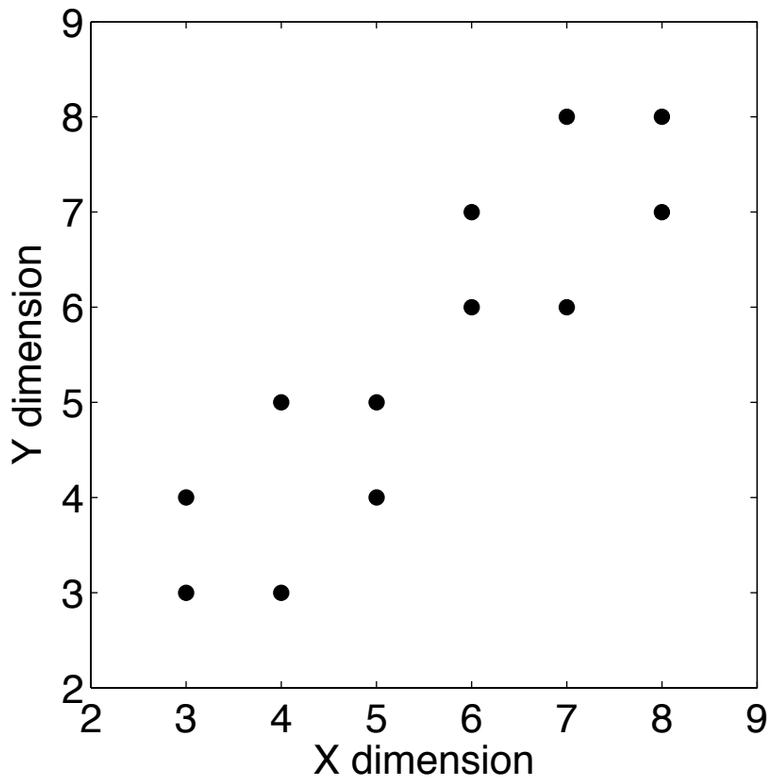
PCA for dimension reduction

$$\vec{x}^{\,\mu} = \begin{pmatrix} x_1^\mu \\ x_2^\mu \\ \dots \\ x_N^\mu \end{pmatrix} \xrightarrow[\text{Rotate via PCA}]{\text{Subtract mean}} \tilde{\vec{x}}^{\,\mu} = \begin{pmatrix} \tilde{x}_1^\mu \\ \tilde{x}_2^\mu \\ \dots \\ \tilde{x}_N^\mu \end{pmatrix} \xrightarrow[\text{First components}]{\text{Keep only}} \begin{pmatrix} \tilde{x}_1^\mu \\ \tilde{x}_2^\mu \end{pmatrix}$$

PCA

Apply the PCA technique on the dataset below in order to reduce its dimensionality.

X	3	3	4	4	5	5	6	6	7	7	8	8
Y	3	4	3	5	4	5	6	7	6	8	7	8



PCA

1. Calculate signal means, subtract means from signals.

X	3	3	4	4	5	5	6	6	7	7	8	8
Y	3	4	3	5	4	5	6	7	6	8	7	8

$$\overline{X} = \sum_{i=1}^{p} \frac{X_i}{p} = \frac{6 * 11}{12} = \frac{11}{2}$$

$$\overline{Y} = \frac{11}{2}$$

PCA

2. Write down the covariance formula. Calculate the covariance matrix.

$$var\left(X\right) = \sum_{i=1}^{p} \frac{\left(X_i - \overline{X}\right)\left(X_i - \overline{X}\right)}{p} = \sum_{i=1}^{p} \frac{\tilde{X}^2}{p} = 35/12$$

$$cov\left(X, Y\right) = \sum_{i=1}^{p} \frac{\left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{p} = \sum_{i=1}^{p} \frac{\tilde{X}\tilde{Y}}{p} = 31/12$$

PCA

2. Write down the covariance formula. Calculate the covariance matrix.

$$C = \begin{pmatrix} var(X) & cov(X,Y) \\ cov(Y,X) & var(Y) \end{pmatrix} = \begin{pmatrix} 35/12 & 31/12 \\ 31/12 & 35/12 \end{pmatrix}$$

PCA

3. Find the eigenvalues and eigenvectors. Make sure that the length of the eigenvectors is 1!

$$det(C - \lambda I) = det \begin{pmatrix} 35/12 - \lambda & 31/12 \\ 31/12 & 35/12 - \lambda \end{pmatrix} = 0$$

$$\lambda_1 = 11/2 \qquad \lambda_2 = 1/3$$

PCA

3. Find the eigenvalues and eigenvectors. Make sure that the length of the eigenvectors is 1!

$$CV_1 = \lambda_1 V_1 \qquad CV_2 = \lambda_2 V_2$$

$$V_1 = \begin{pmatrix} \sqrt{2}/2 \\ \sqrt{2}/2 \end{pmatrix} \qquad V_2 = \begin{pmatrix} -\sqrt{2}/2 \\ \sqrt{2}/2 \end{pmatrix}$$

PCA

4. Decide how many eigenvectors you keep. From the eigenvectors form the “Feature Vector”.

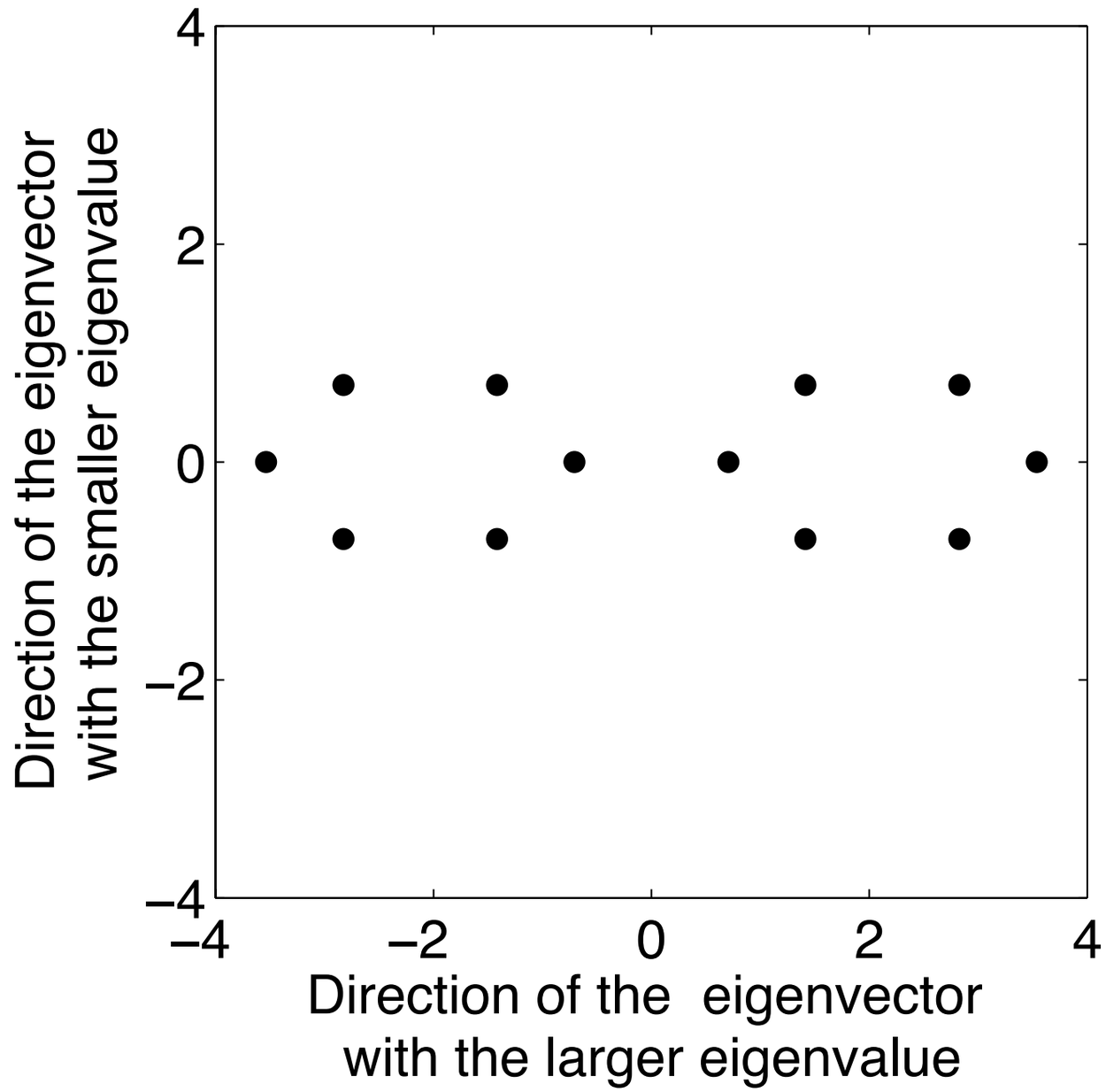
$$F = \begin{pmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{pmatrix}$$

5. Derive the new data:
FinalData=RowFeatureVectorxRowDataAdjust.

RowFeatureVector has the eigenvectors in the rows with the eigenvector with the highest eigenvalue On TOP.
RowDataAdjust contains the zero mean data in each column, with each row holding a separate dimension.

$$D_n = F^T D_c$$

PCA


4
2
0
−2
−4
−4
−2
0
2
4
Direction of the eigenvector with the smaller eigenvalue
Direction of the eigenvector with the larger eigenvalue

PCA

4. Decide how many eigenvectors you keep. From the eigenvectors form the “Feature Vector”.

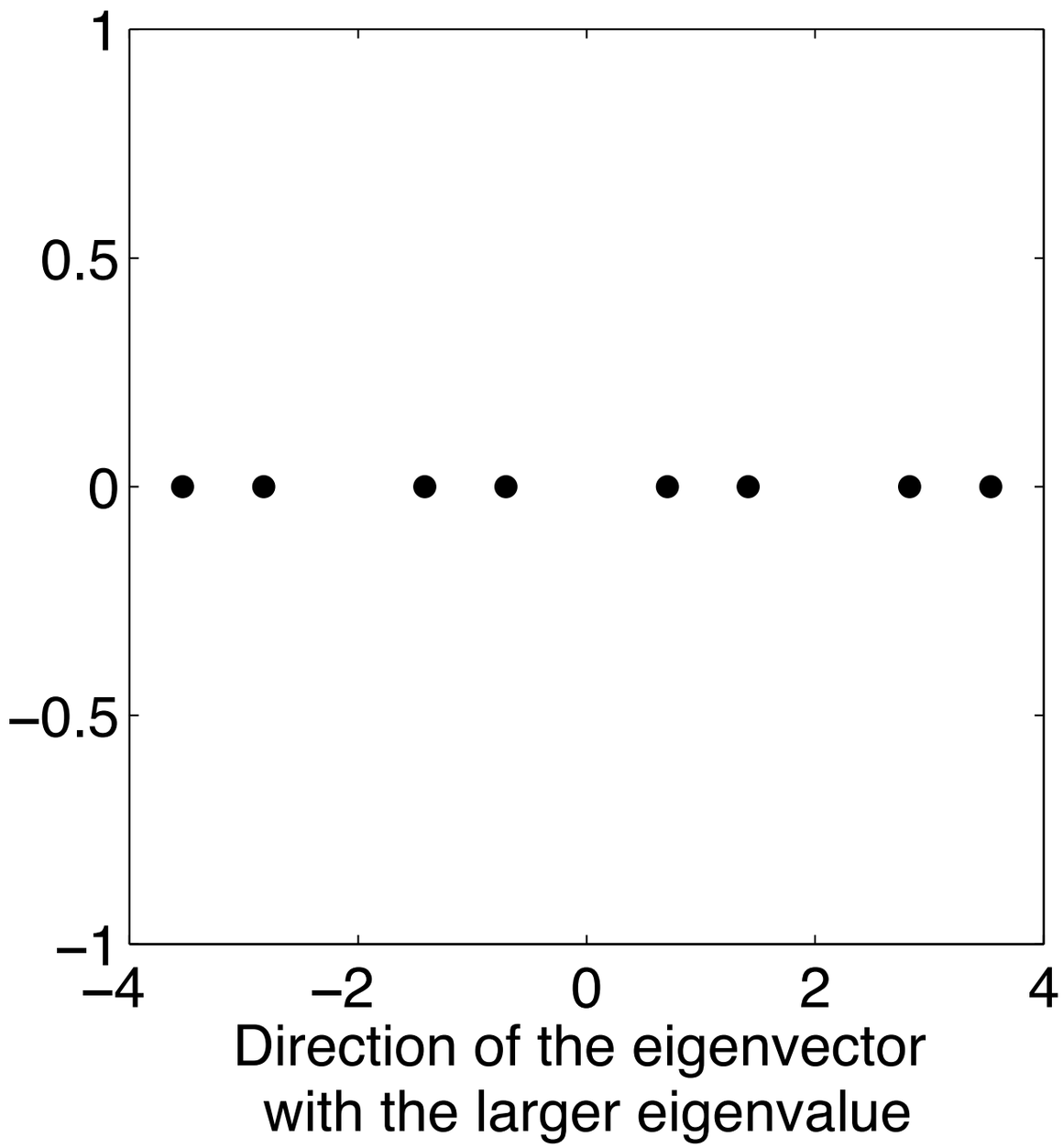
$$F^T = \begin{pmatrix} \sqrt{2}/2 & \sqrt{2}/2 \end{pmatrix}$$

5. Derive the new data:
FinalData=RowFeatureVectorxRowDataAdjust.

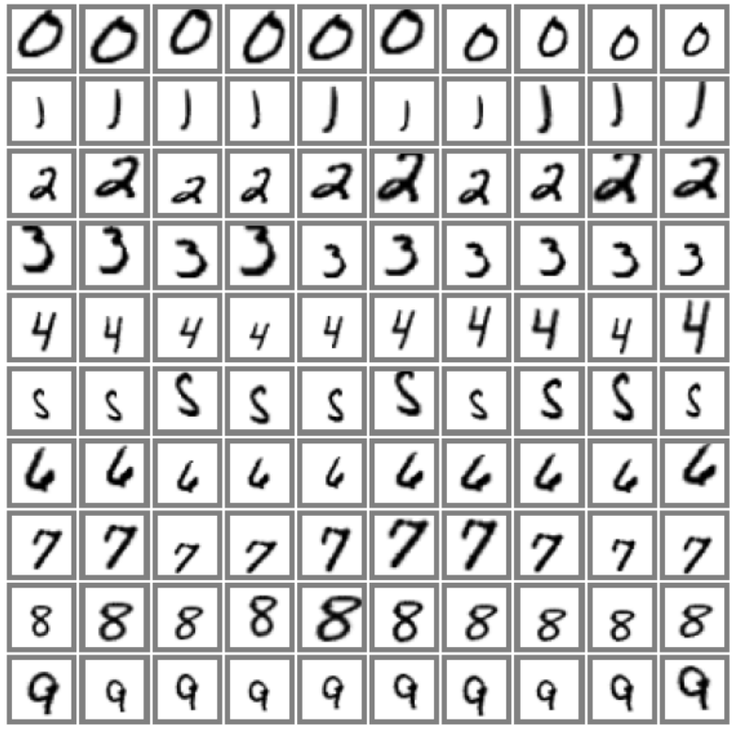
RowFeatureVector has the eigenvectors in the rows with the eigenvector with the highest eigenvalue On TOP.
RowDataAdjust contains the zero mean data in each column, with each row holding a separate dimension.

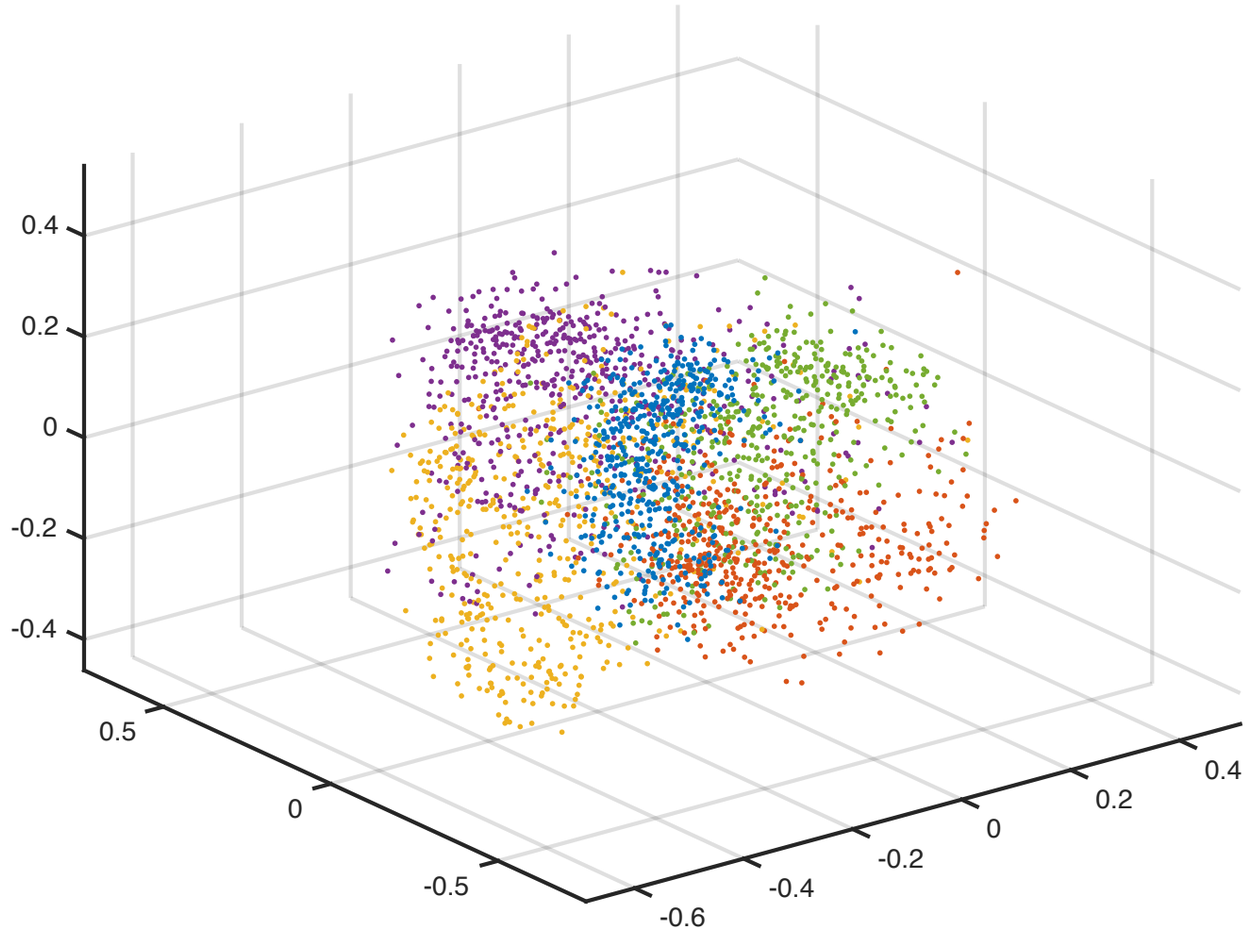
$$D_n = F^T D_c$$

PCA

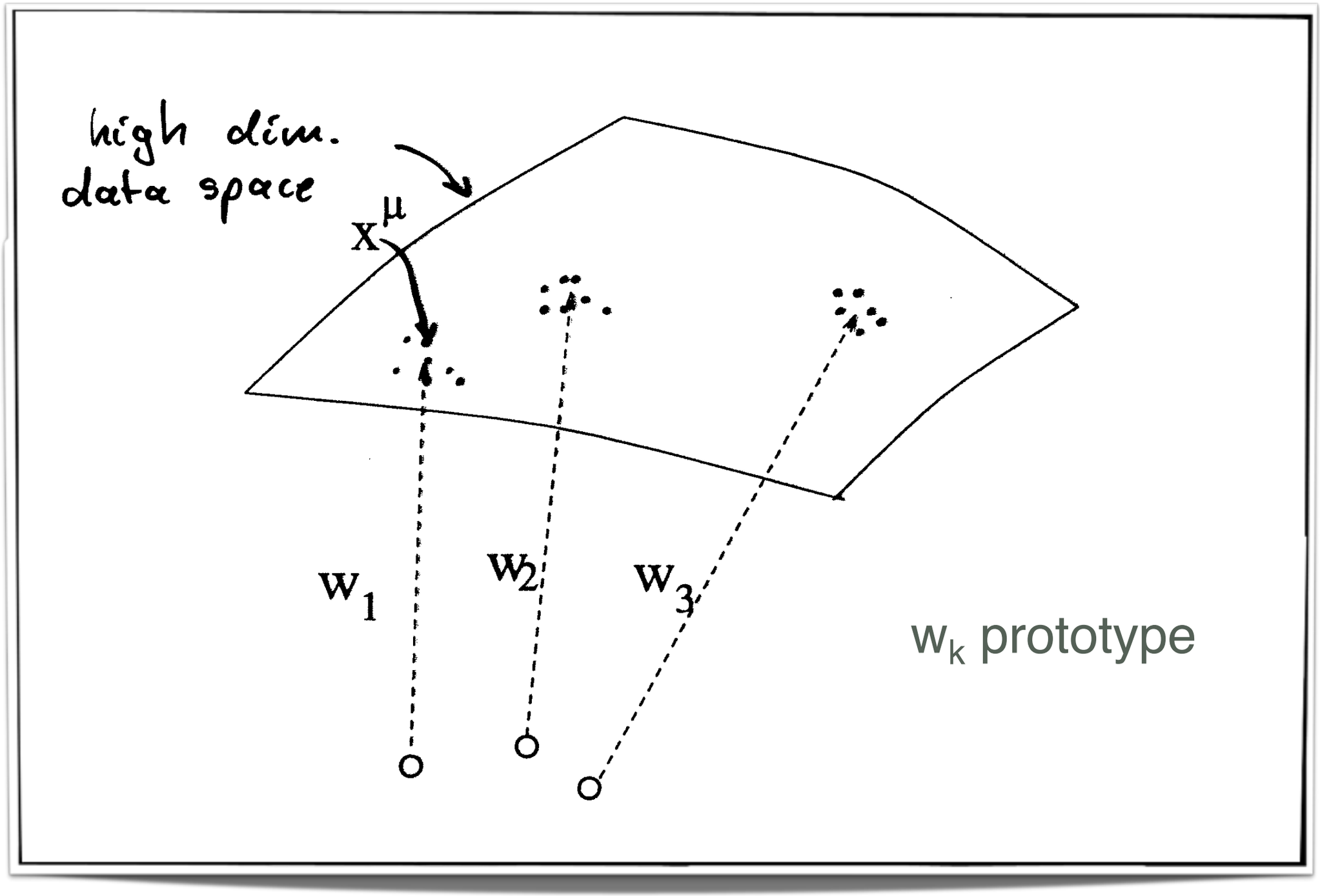

1
0.5
0
−0.5
−1
−4
−2
0
2
4
Direction of the eigenvector
with the larger eigenvalue

PCA





Unsupervised Learning



Summary

- When using continuous notation, dynamical systems methods can be applied for analysing learning rules.
  - E.g. Theory of Oja's rule.
- PCA can be used for cluster visualisation.

Thank you!

Acknowledgements

- This module is an adaptation of the MSc module “Unsupervised and Reinforcement Learning in Neural Networks” by Prof. Wulfram Gerstner at EPFL, Switzerland.

Bibliography

- Neuronal Dynamics, by Gerstner et al.
- Introduction to the Theory of Neural Computation" by Hertz, Krogh & Palmer
- Reinforcement Learning: An Introduction" by Sutton & Barto
- Scholarpedia