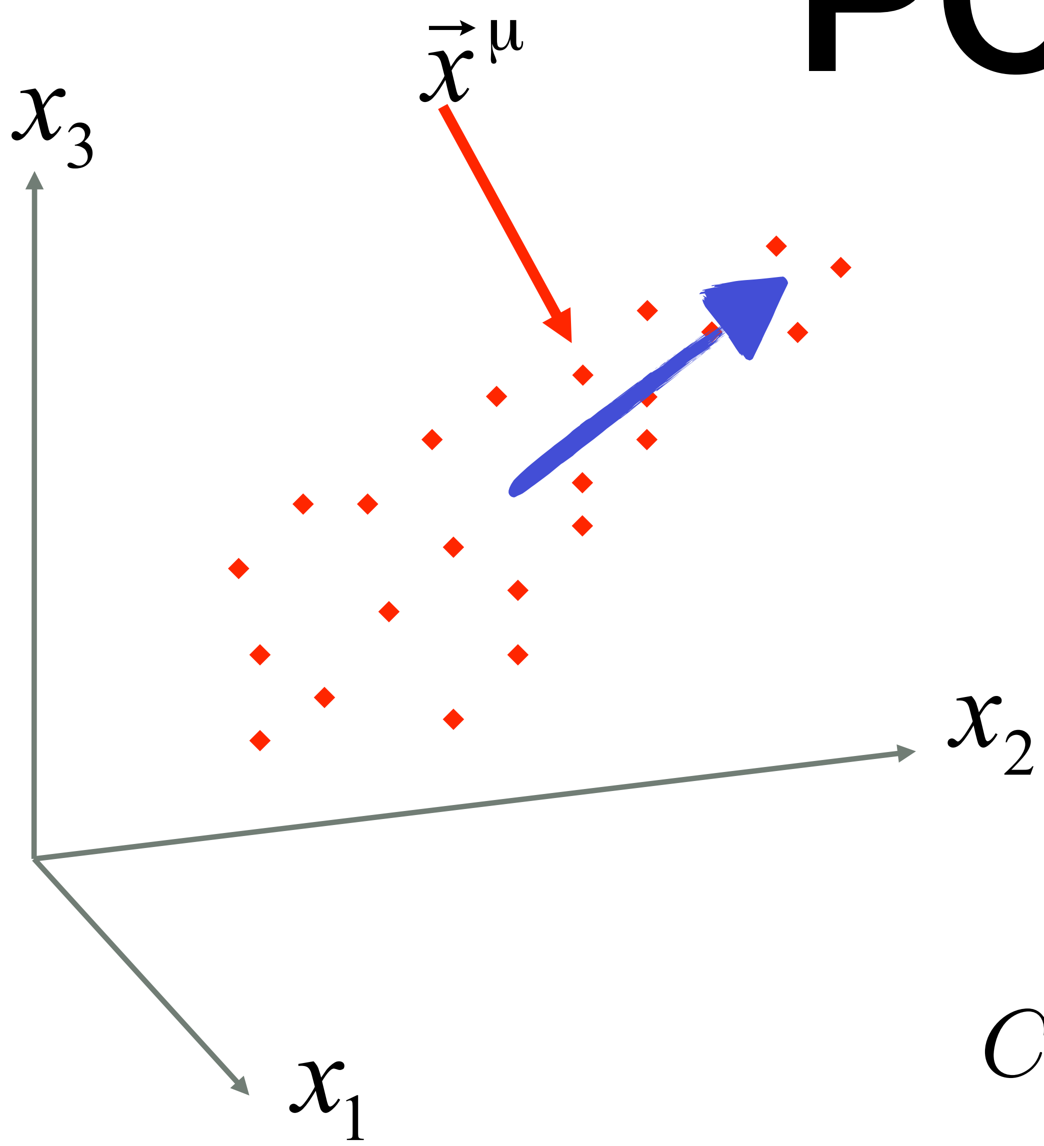
Adaptive Intelligence

Prof. Eleni Vasilaki

PCA



Correlation:

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

Covariance:

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

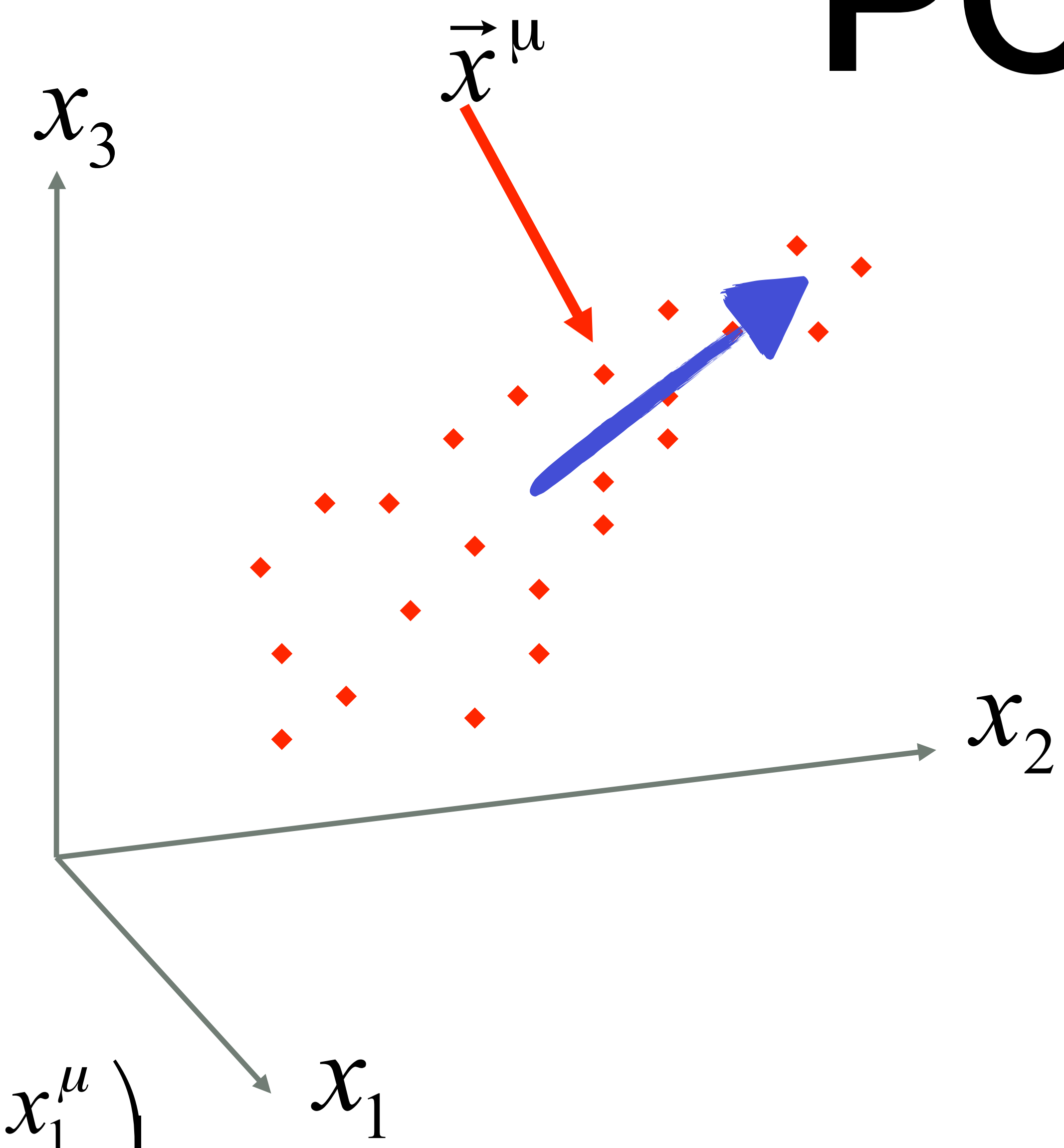
$$\mathbf{C^0 e_g{}^n} = \lambda^{\mathbf{n}} \mathbf{e_g{}^n} \qquad \lambda^1 > \lambda^2 > \ldots > \lambda^N$$

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

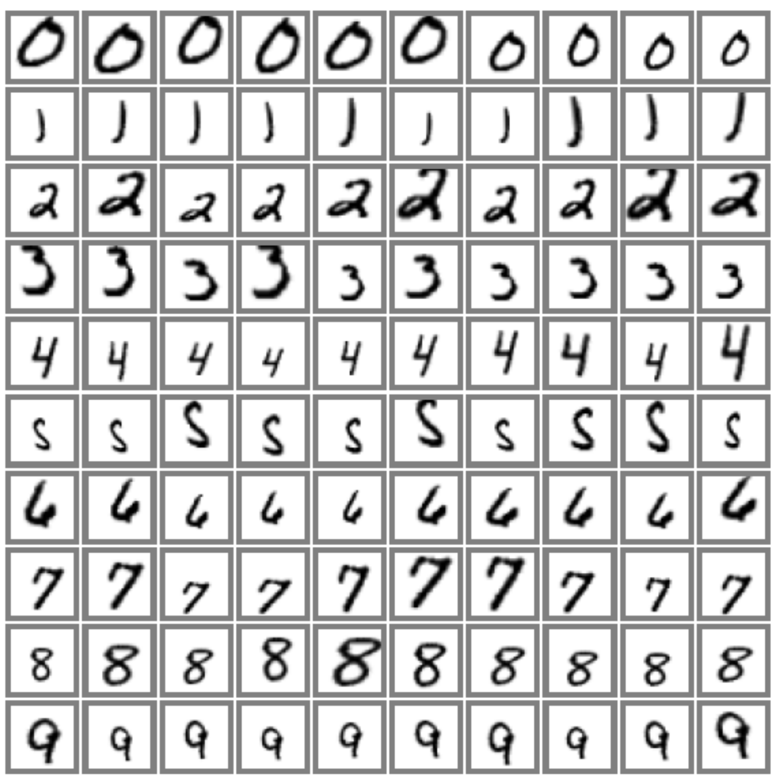


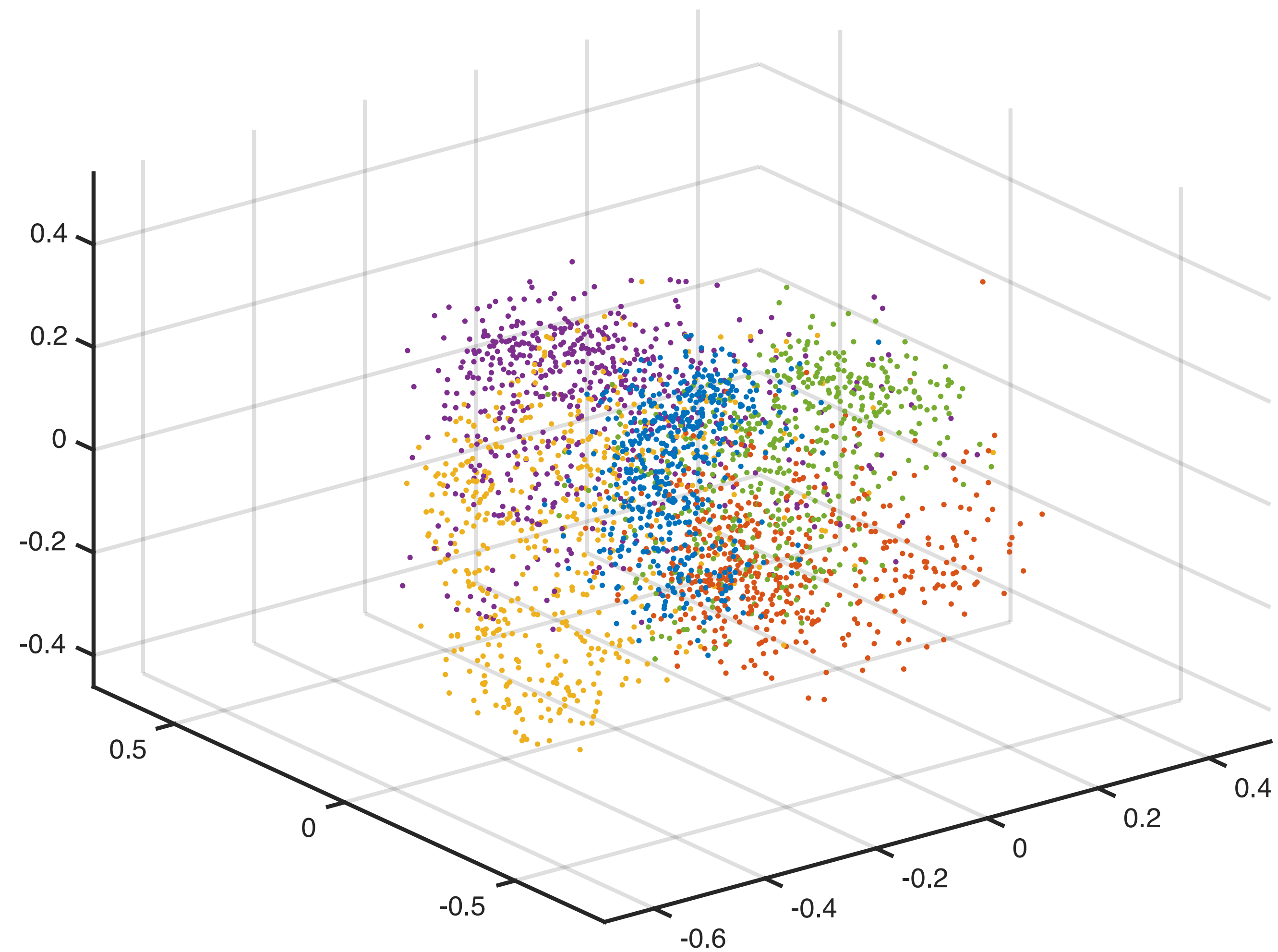
PCA Tutorial:
Lindsay I Smith

PCA for dimension reduction

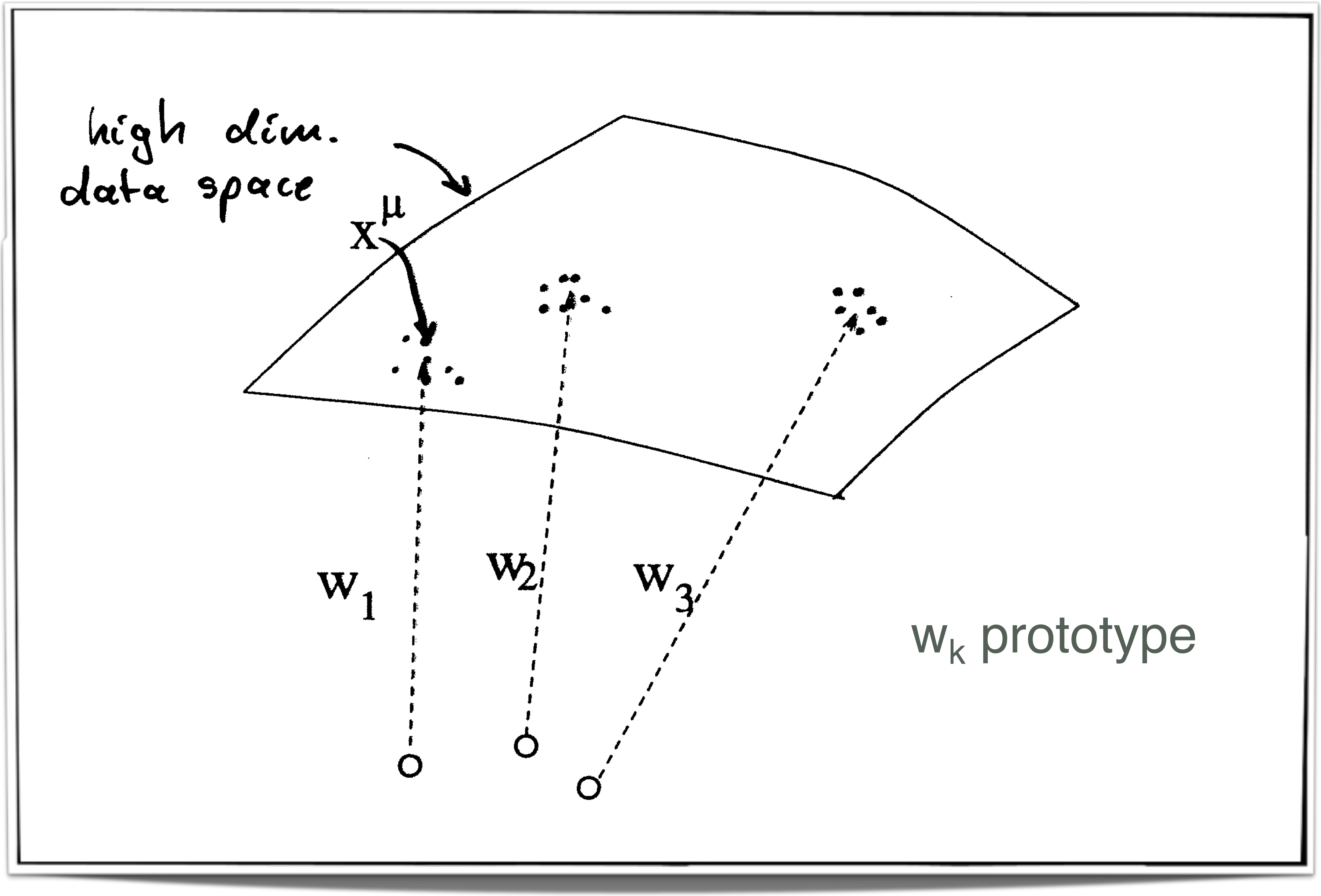
$$\vec{x}^{\mu} = \begin{pmatrix} x_1^{\mu} \\ x_2^{\mu} \\ \dots \\ x_N^{\mu} \end{pmatrix} \xrightarrow[\text{Rotate via PCA}]{\text{Subtract mean}} \tilde{\vec{x}}^{\mu} = \begin{pmatrix} \tilde{x}_1^{\mu} \\ \tilde{x}_2^{\mu} \\ \dots \\ \tilde{x}_N^{\mu} \end{pmatrix} \xrightarrow[\text{First components}]{\text{Keep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

PCA





Unsupervised Learning

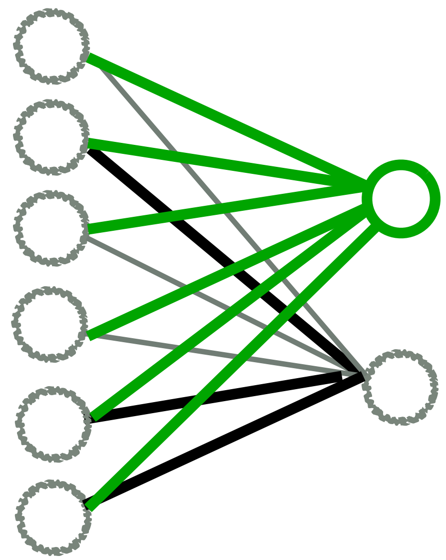


Hebbian Rules

- Minimalistic Hebb rule
- BCM rule (experimental evidence)
- Oja’s rule (Artificial Neural Networks)
    - Link to PCA

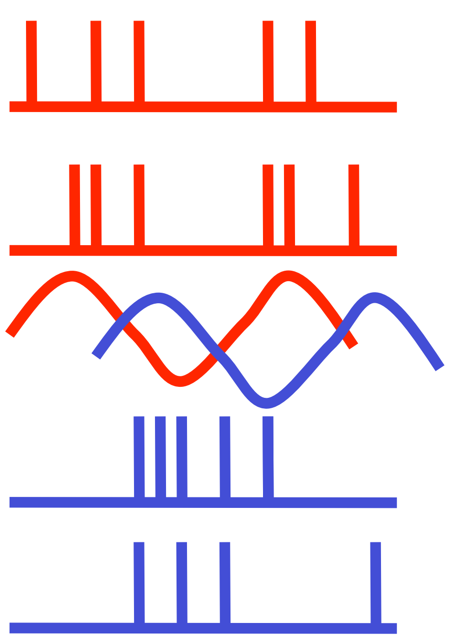
Receptive field development

Initial: random connections



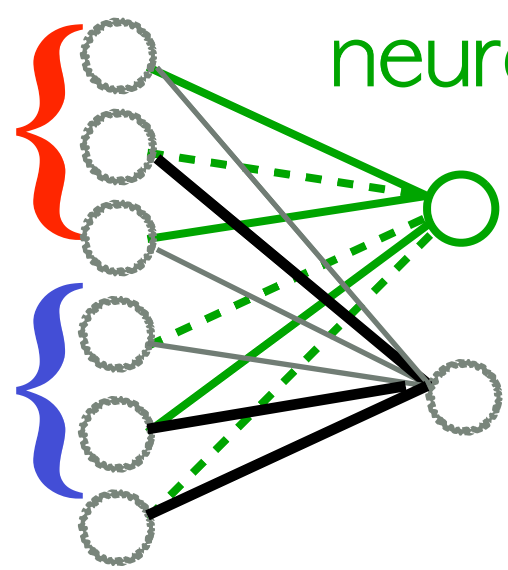
unselective neurons

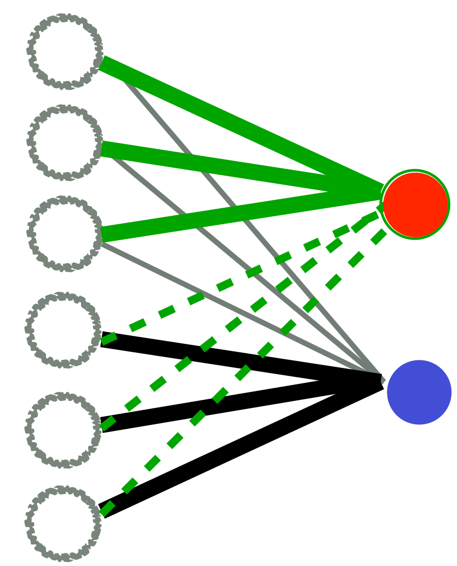
Hubel and Wiesel



Correlated input

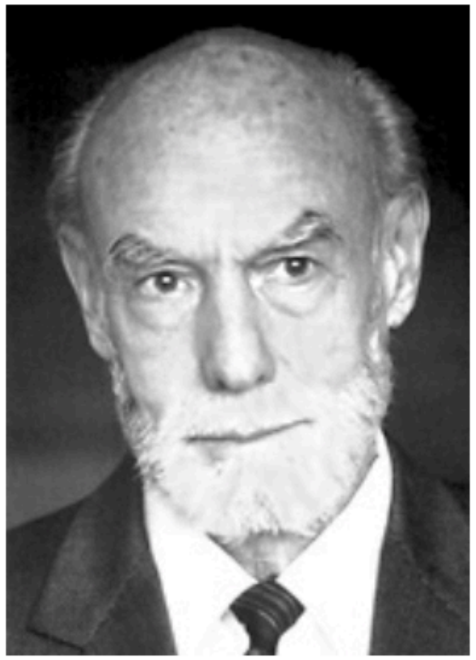
output neurons



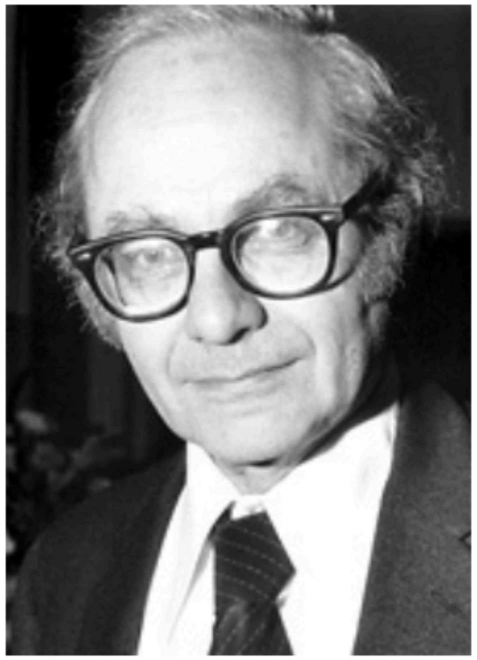


output neurons specialize

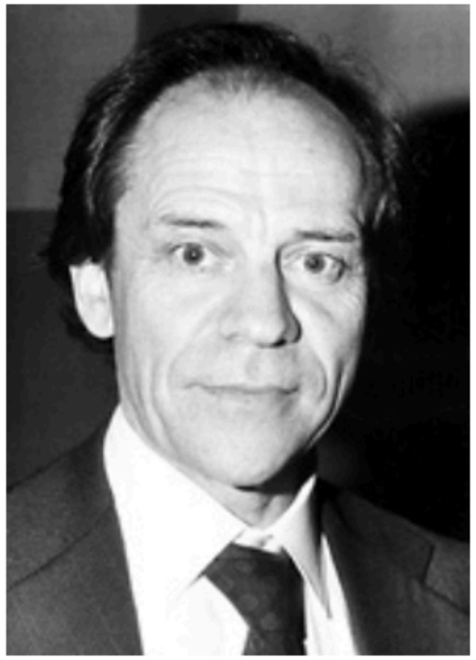
The Nobel Prize in Physiology or Medicine 1981



Roger W. Sperry
Prize share: 1/2



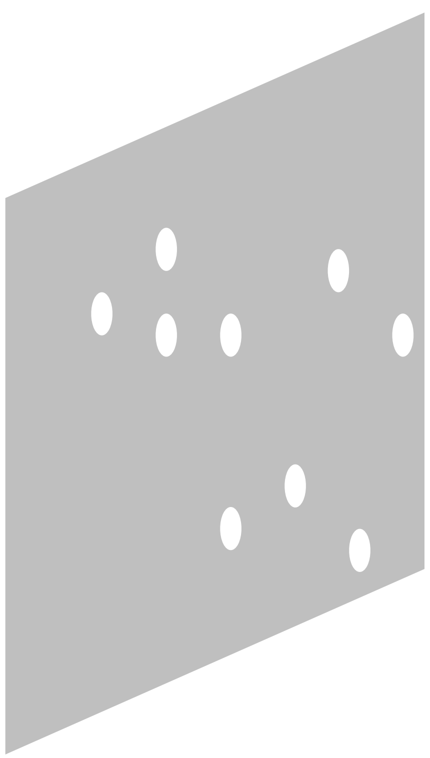
David H. Hubel
Prize share: 1/4

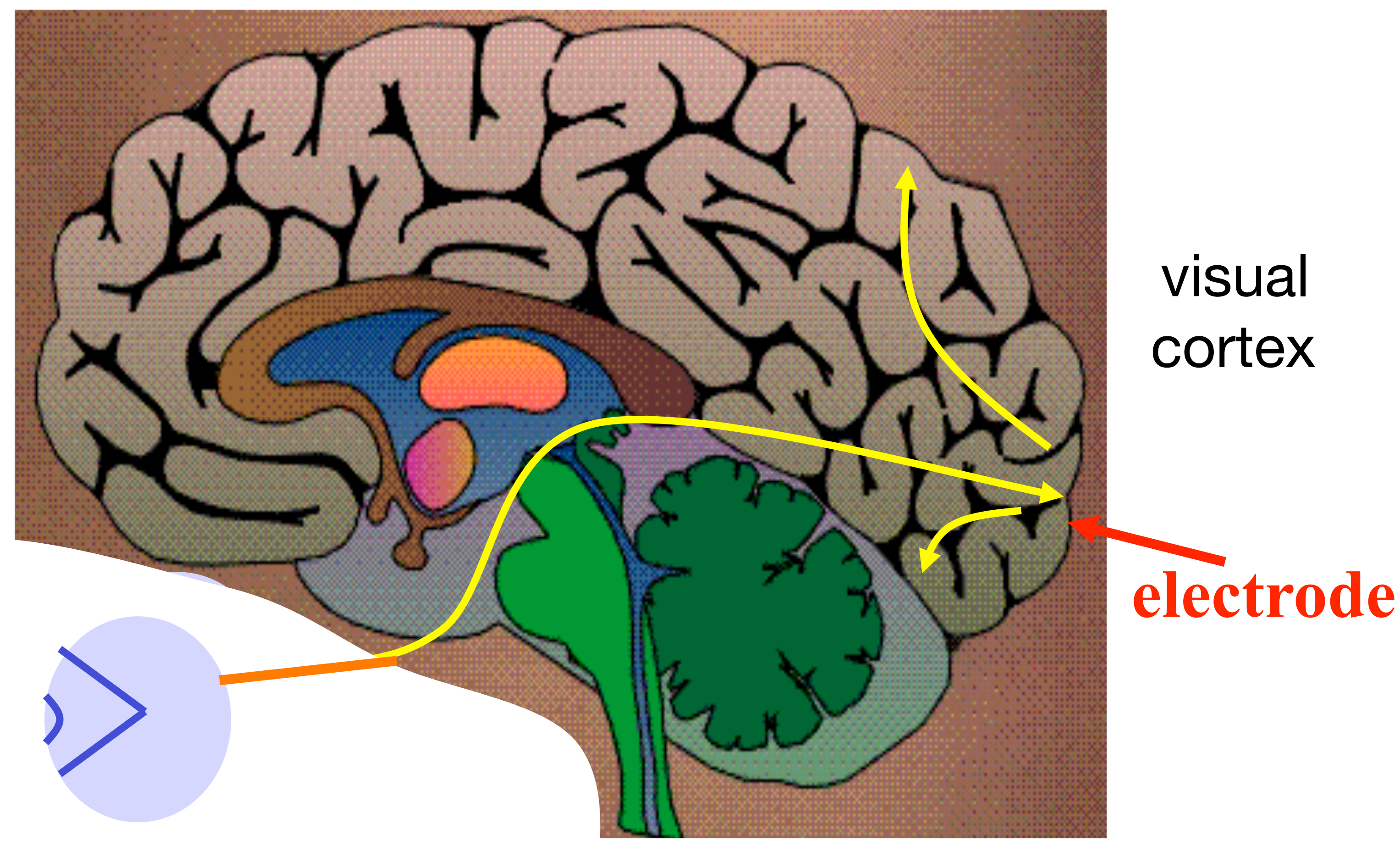


Torsten N. Wiesel
Prize share: 1/4

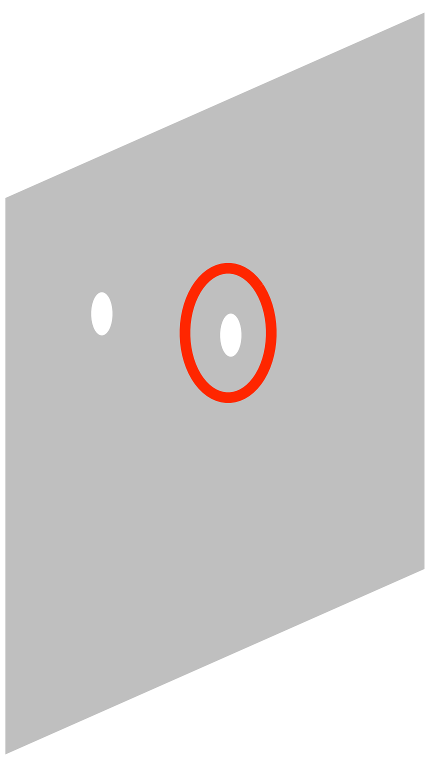
The Nobel Prize in Physiology or Medicine 1981 was divided, one half awarded to Roger W. Sperry *"for his discoveries concerning the functional specialization of the cerebral hemispheres"*, the other half jointly to David H. Hubel and Torsten N. Wiesel *"for their discoveries concerning information processing in the visual system"*.

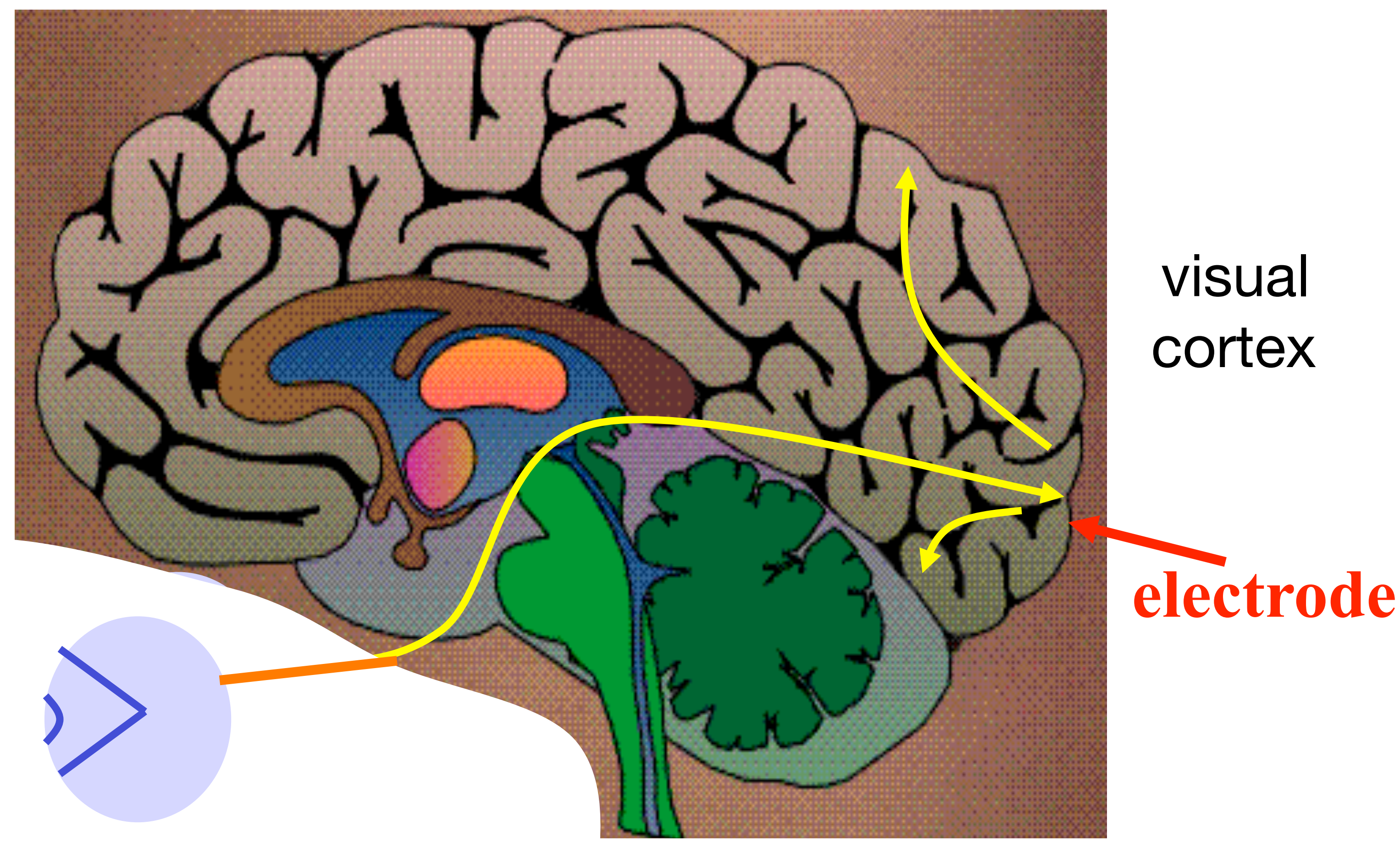
Receptive field development





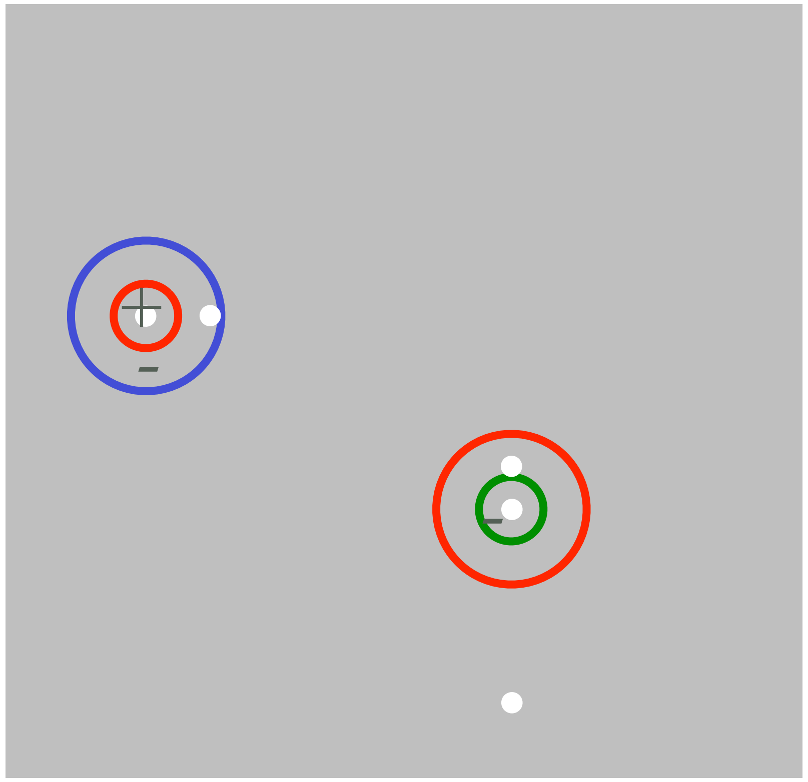
Receptive field development

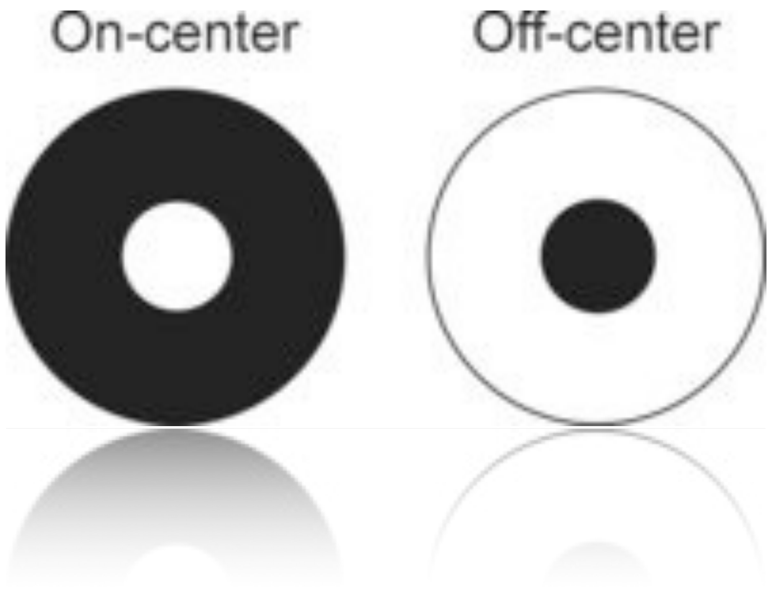




Receptive field development

Receptive fields:
Retina, LGN

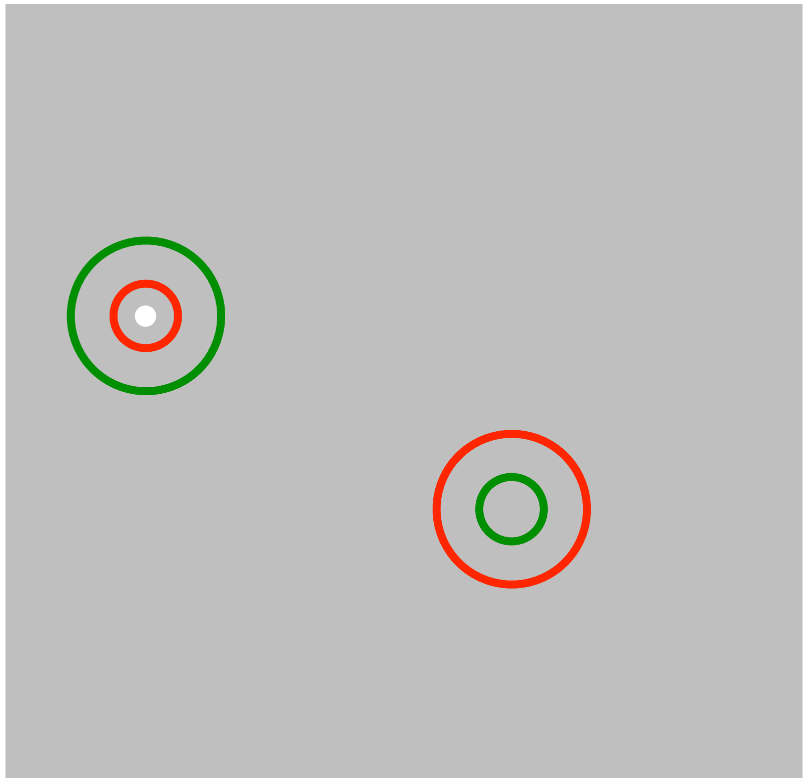




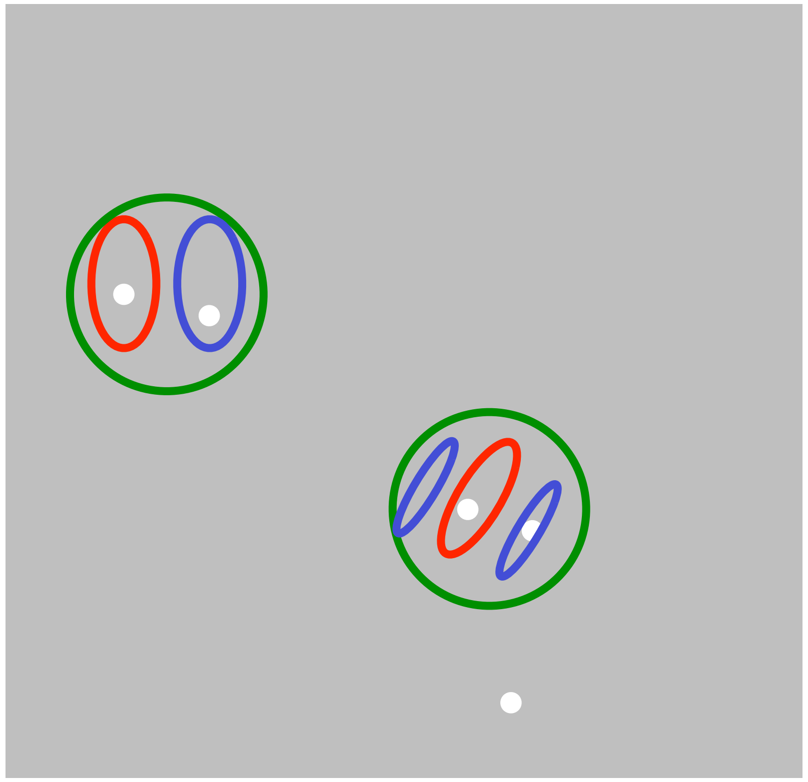
lateral geniculate nucleus (LGN)

Receptive field development

Receptive fields:
Retina, LGN



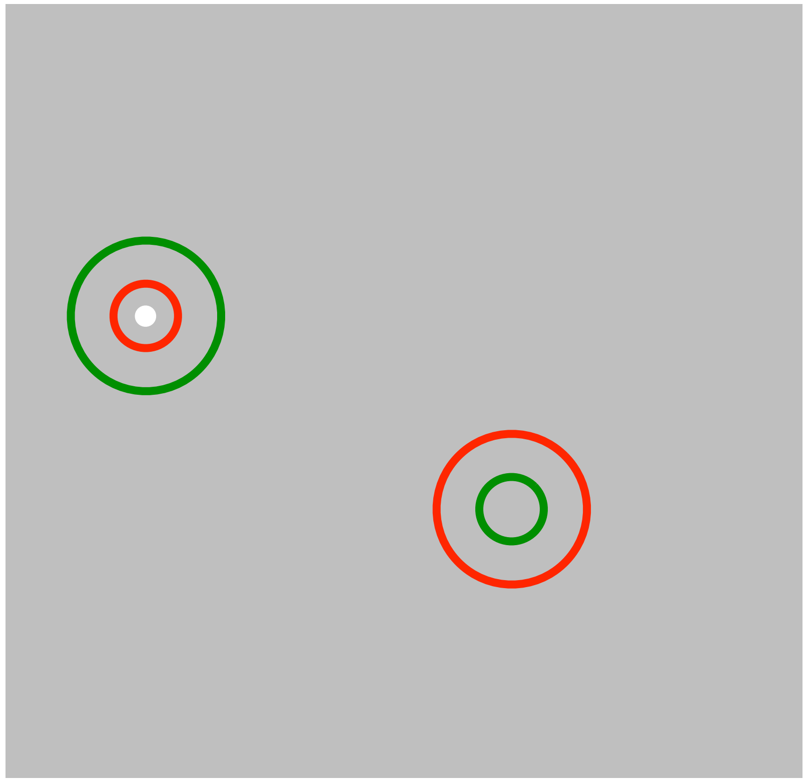
Receptive fields:
visual cortex V1



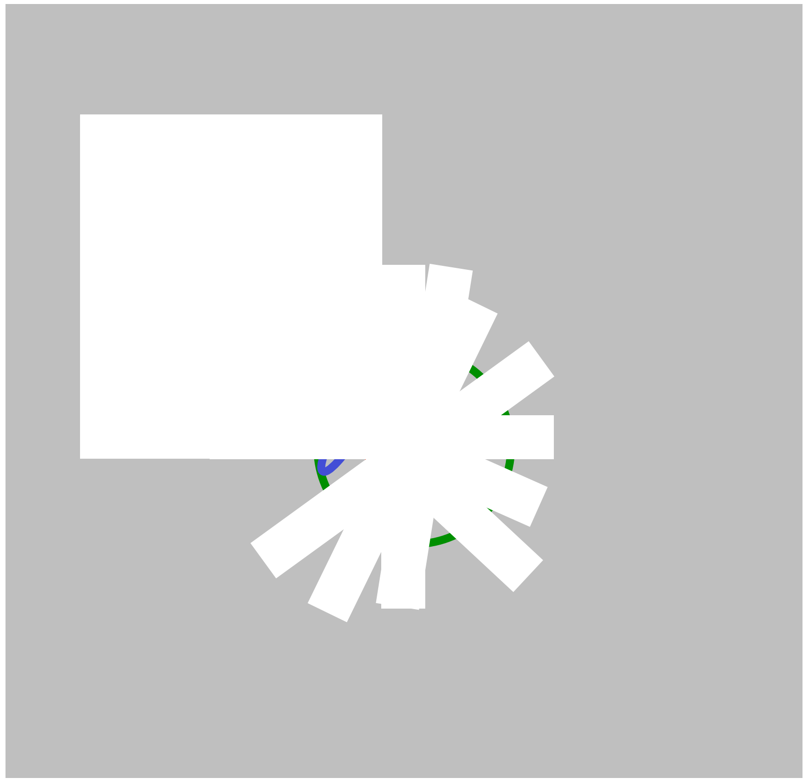
Orientation selective

Receptive field development

Receptive fields:
Retina, LGN

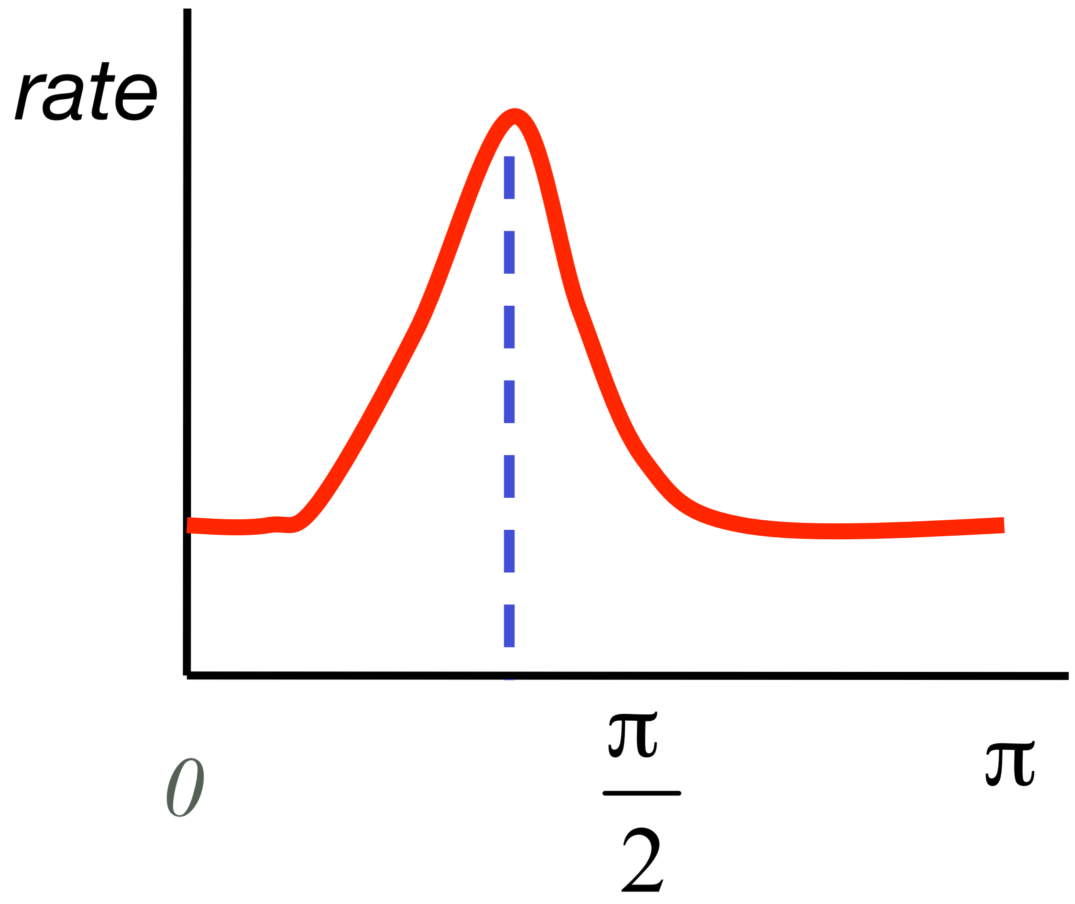


Receptive fields:
visual cortex V1



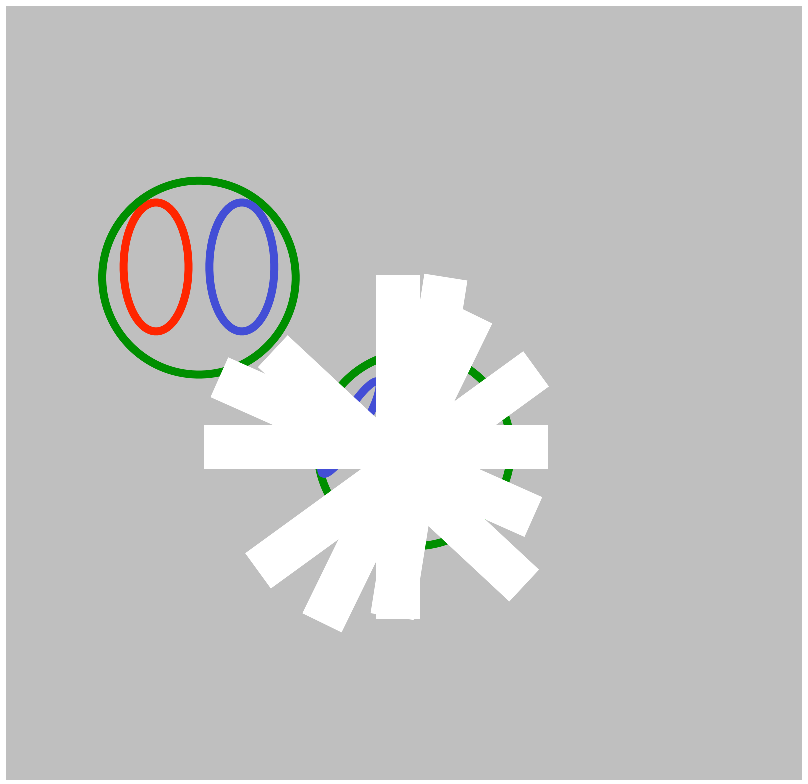
Orientation selective

Receptive field development



Stimulus orientation

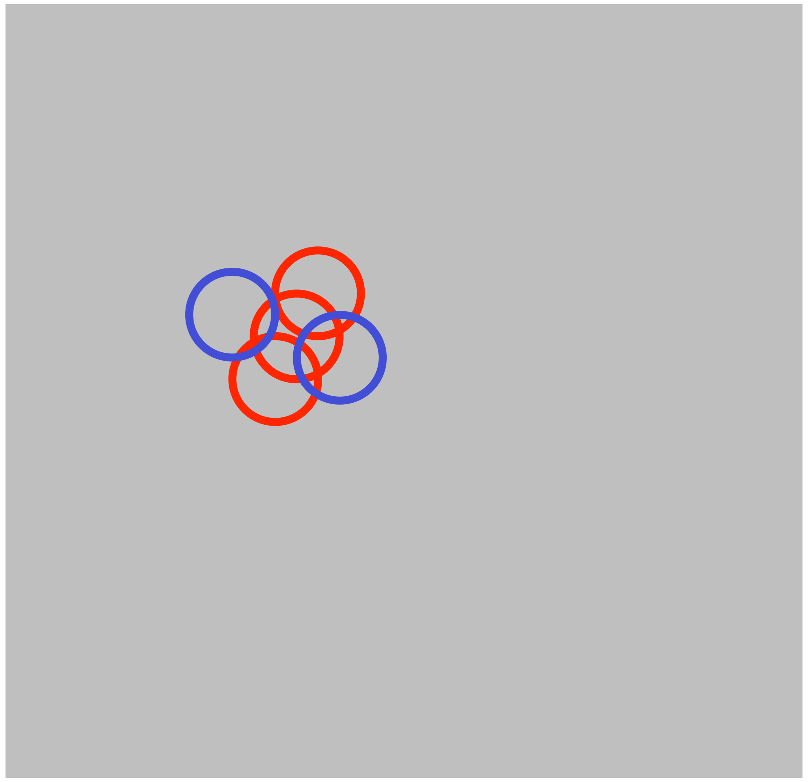
Receptive fields:
visual cortex V1



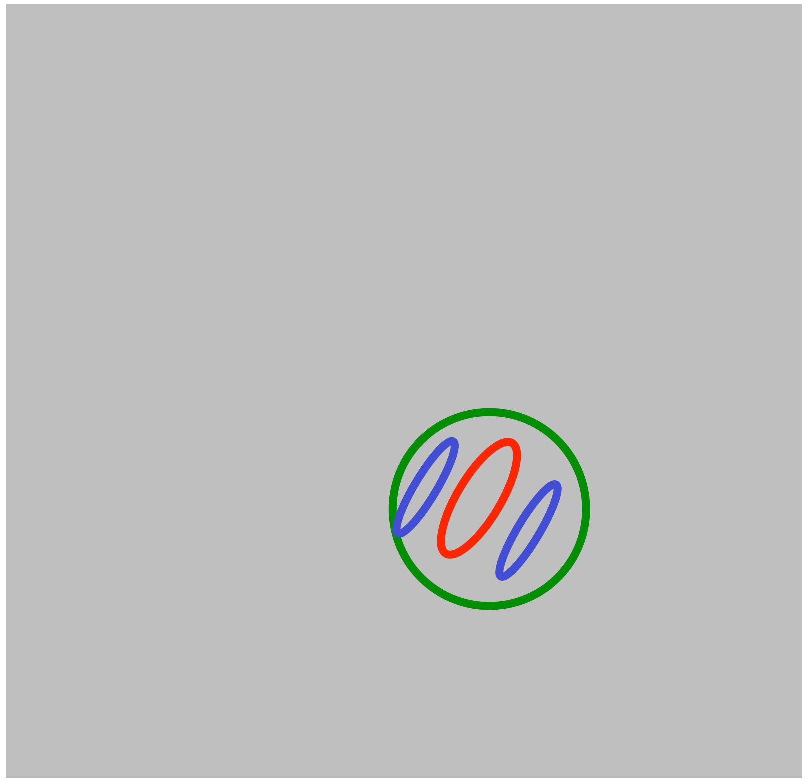
Orientation selective

Receptive field development

Receptive fields:
Retina, LGN



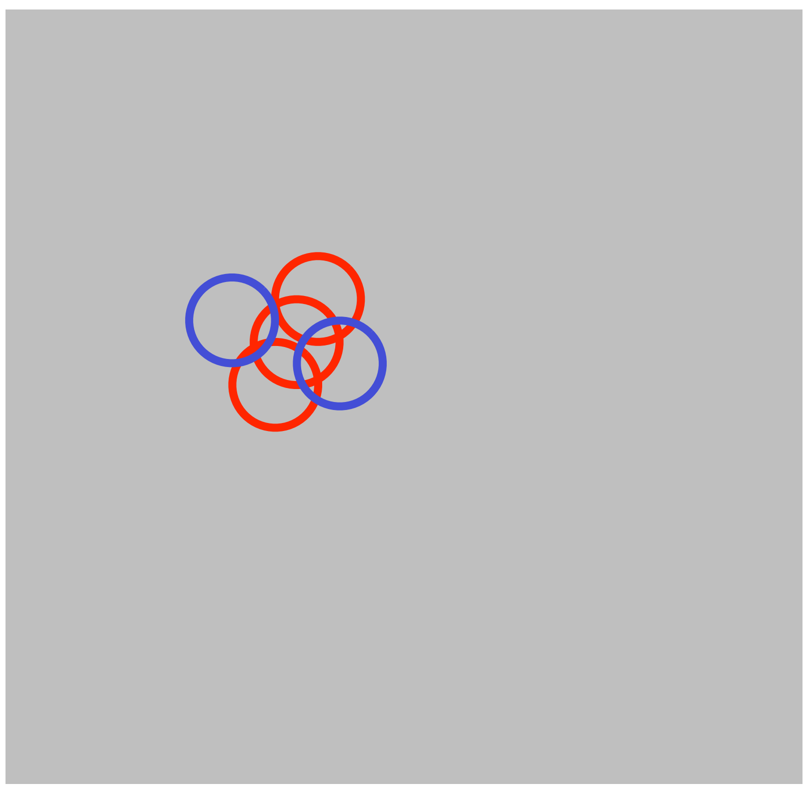
Receptive fields:
visual cortex V1



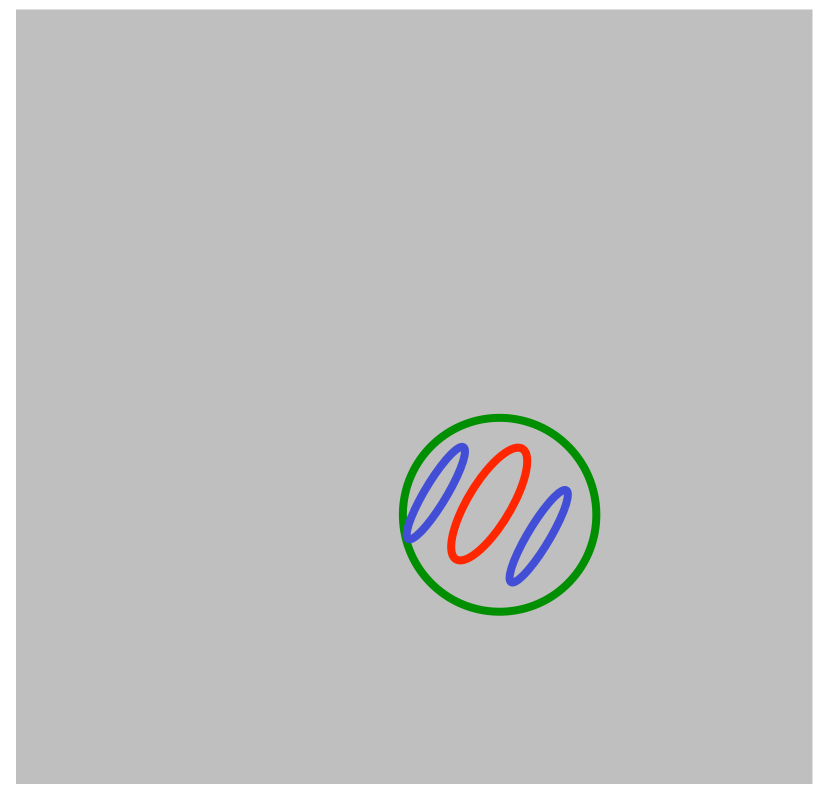
What makes cells orientation selective ?

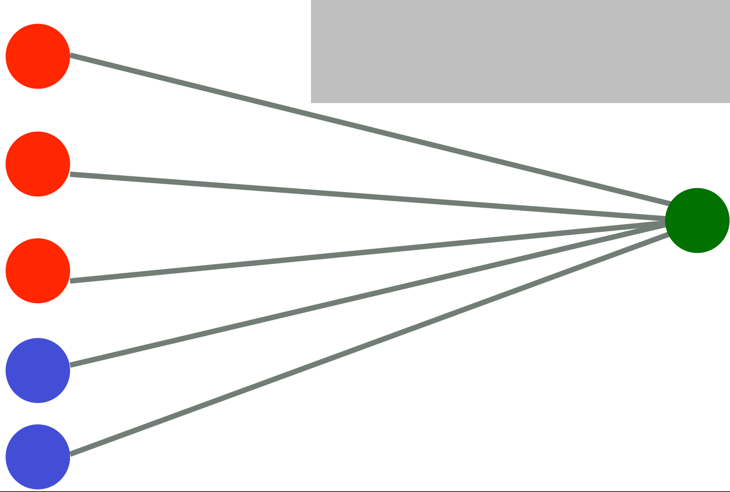
Receptive field development

Receptive fields:
Retina, LGN



Receptive fields:
visual cortex V1

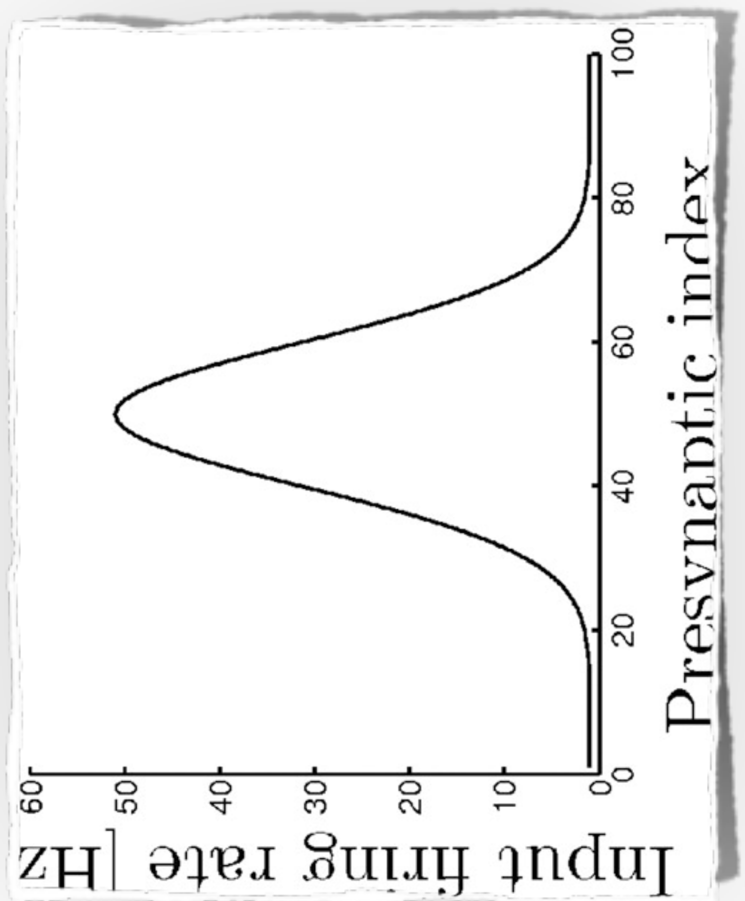


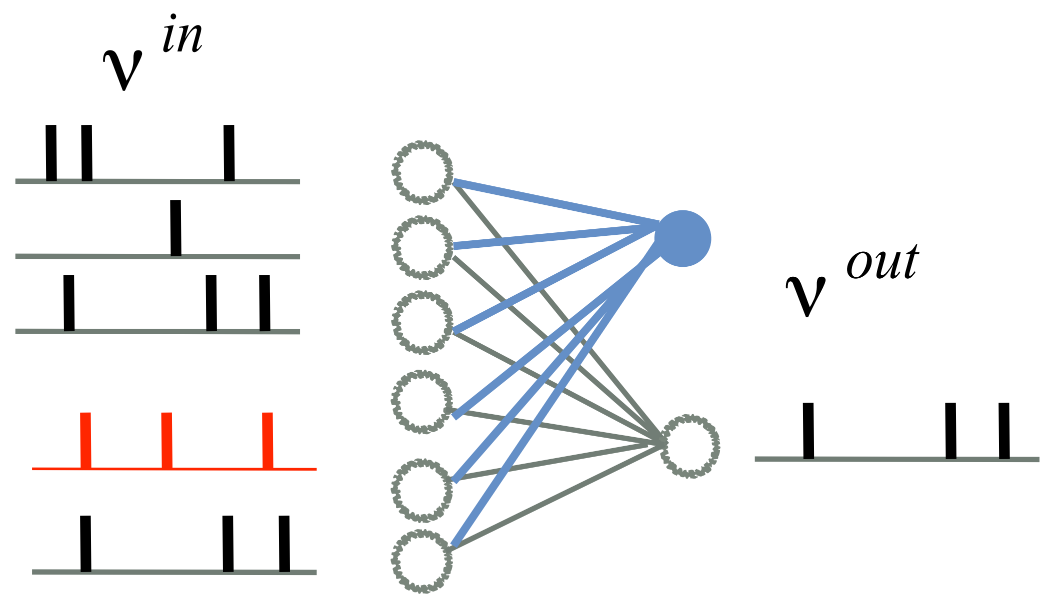


Precise wiring. How is this achieved ?

Hebbian learning.

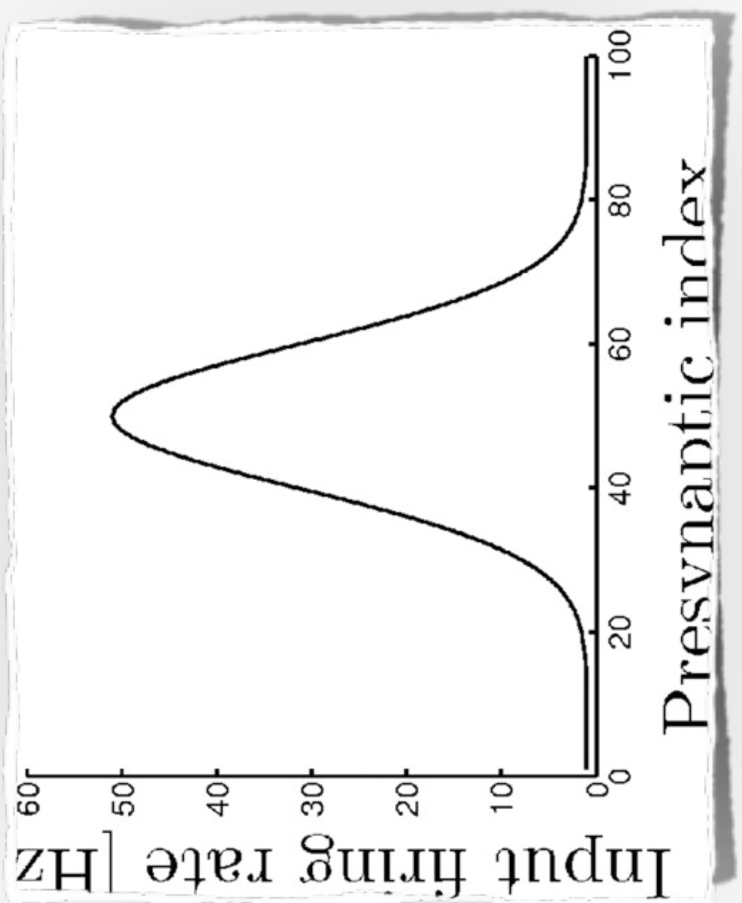
Receptive field development

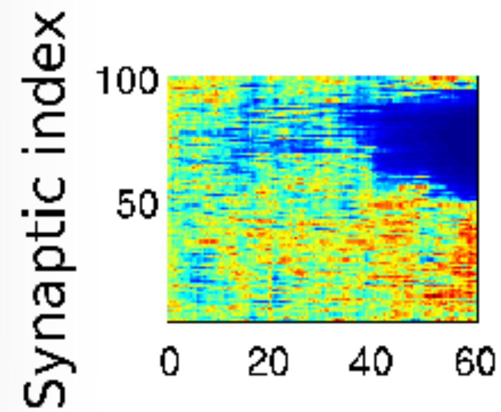




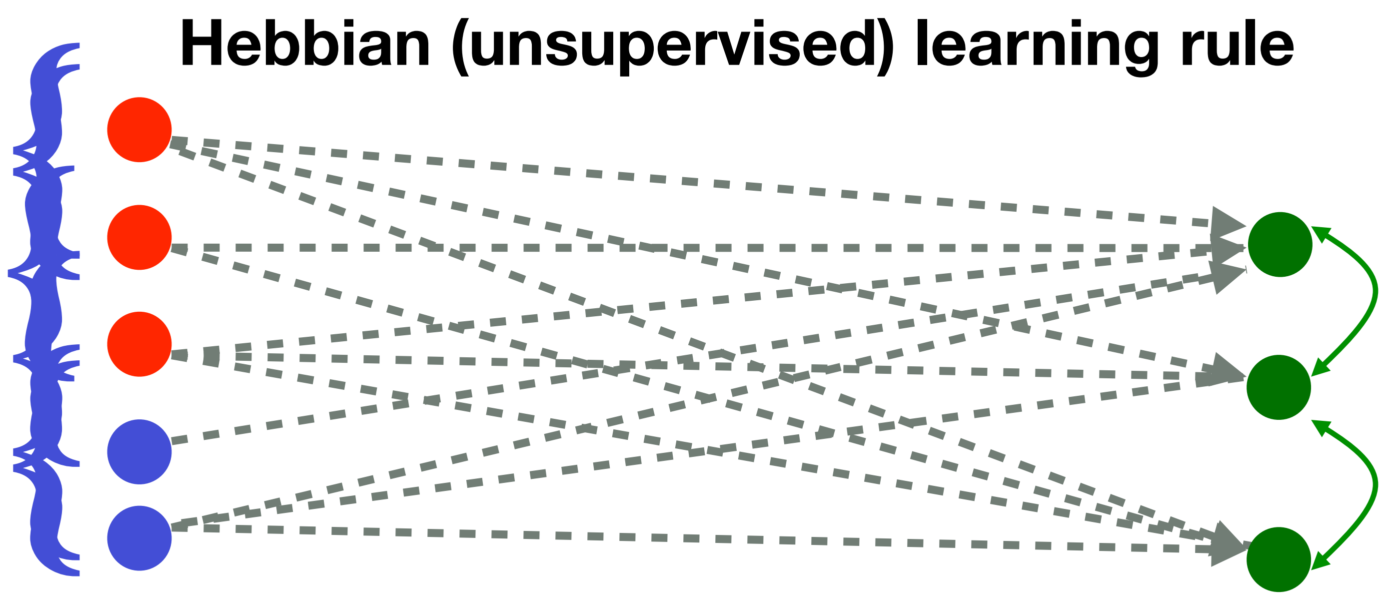
Spatially structured input

Receptive field development





Receptive field development

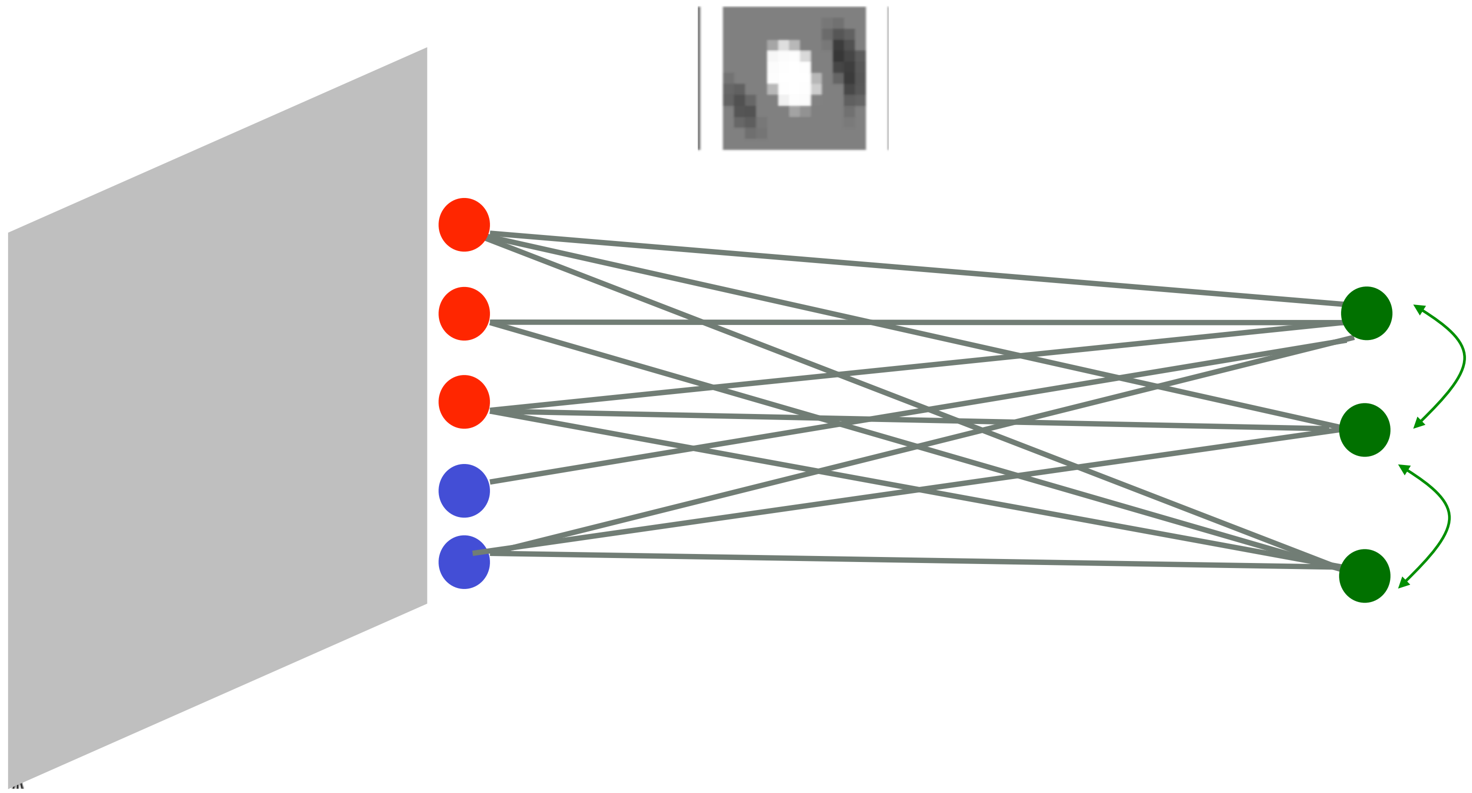


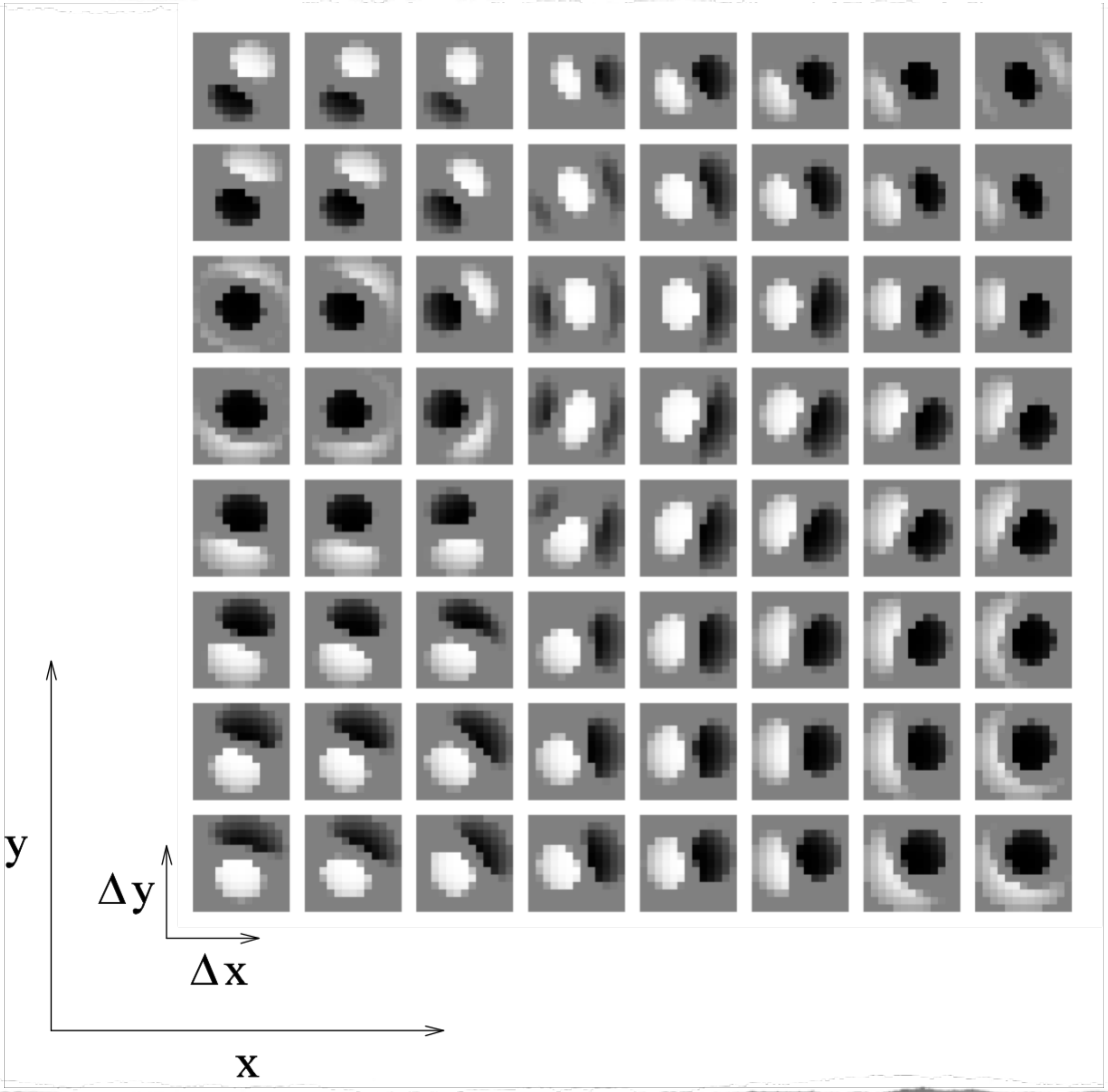
LGN neurons

cortical neurons

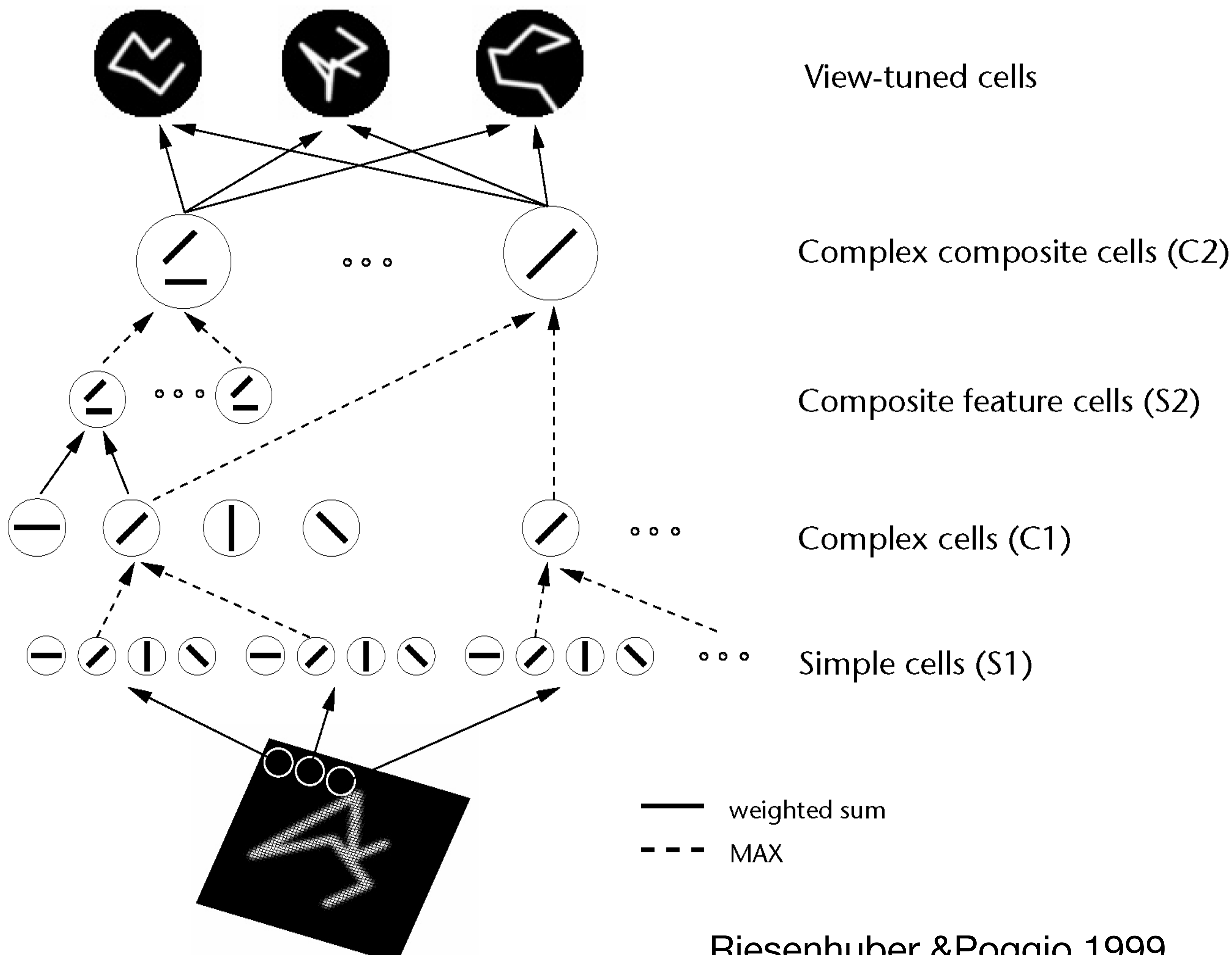
Stimulation with
Locally correlated inputs

Wiring develops=
receptive fields development



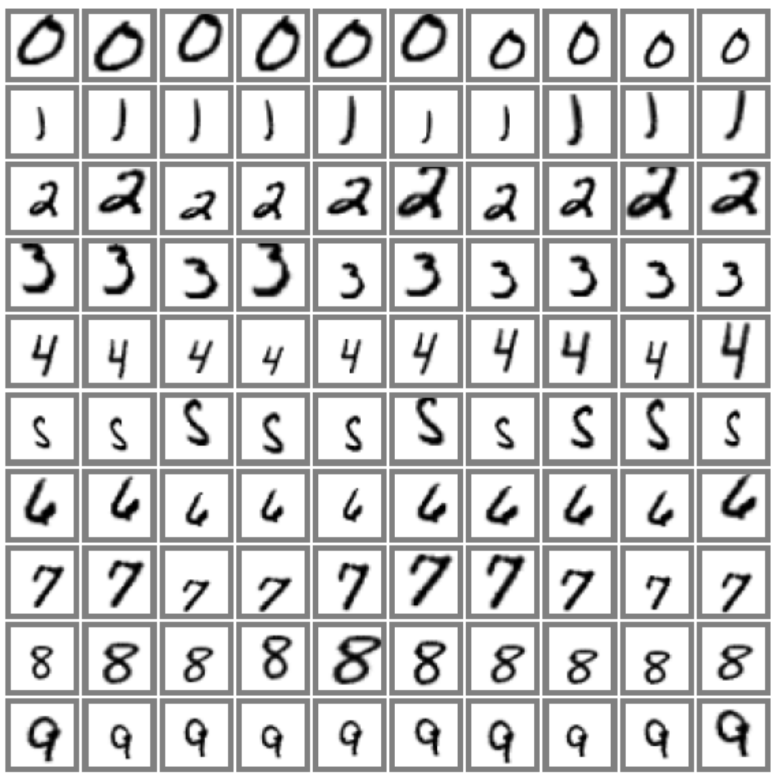

y
Δy
Δx
x

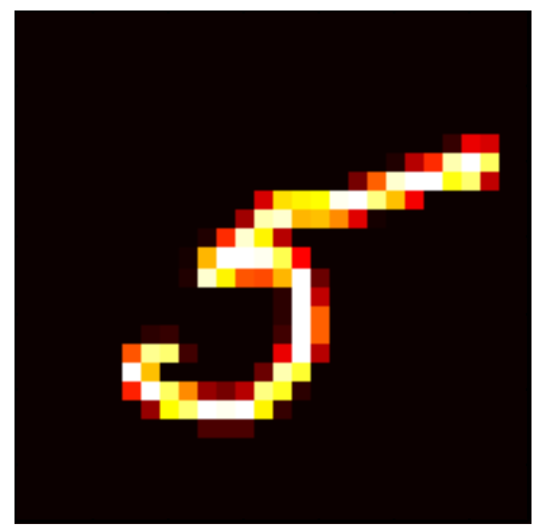
Selectivity and Competition



Riesenhuber &Poggio 1999

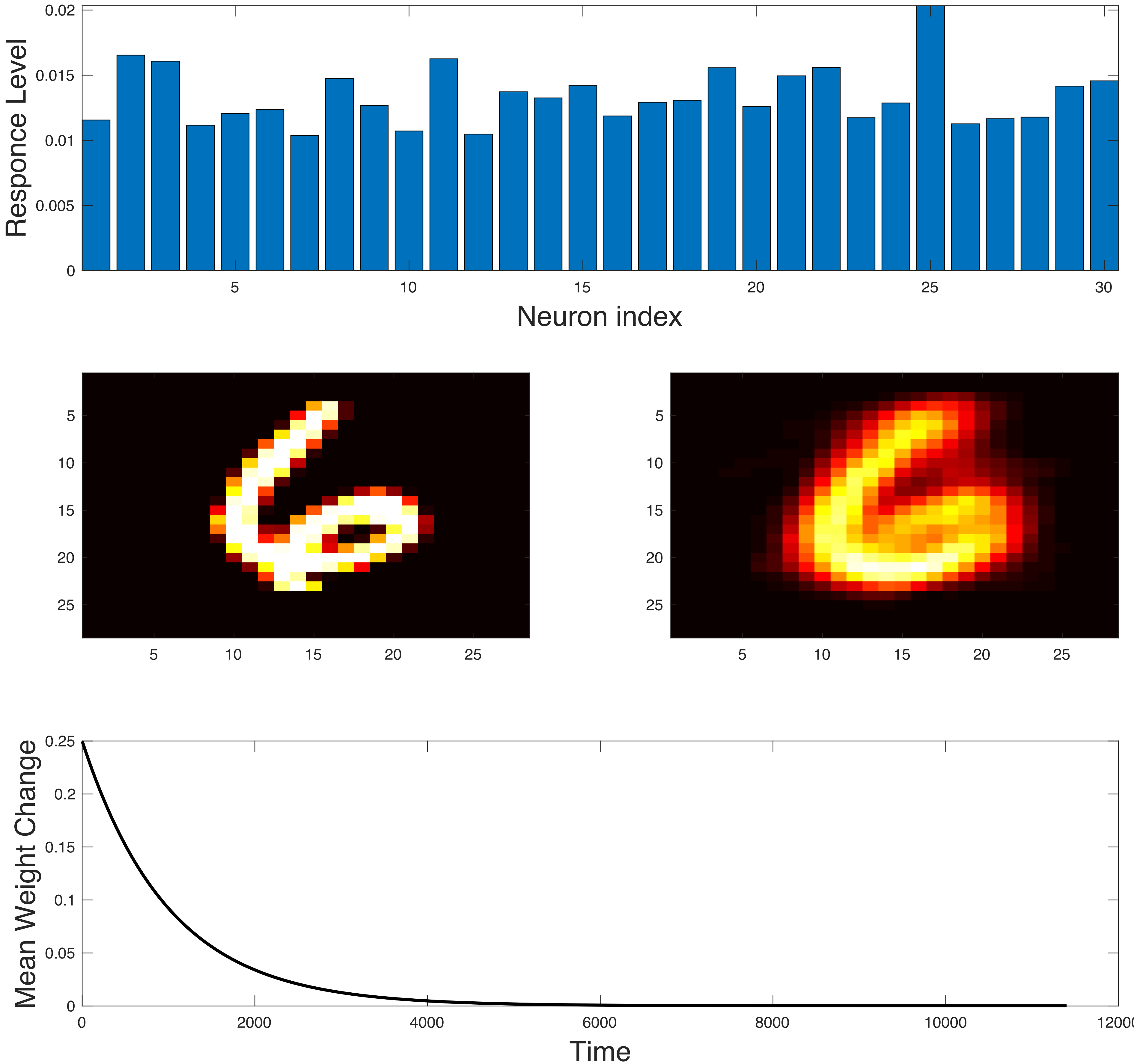
Competition



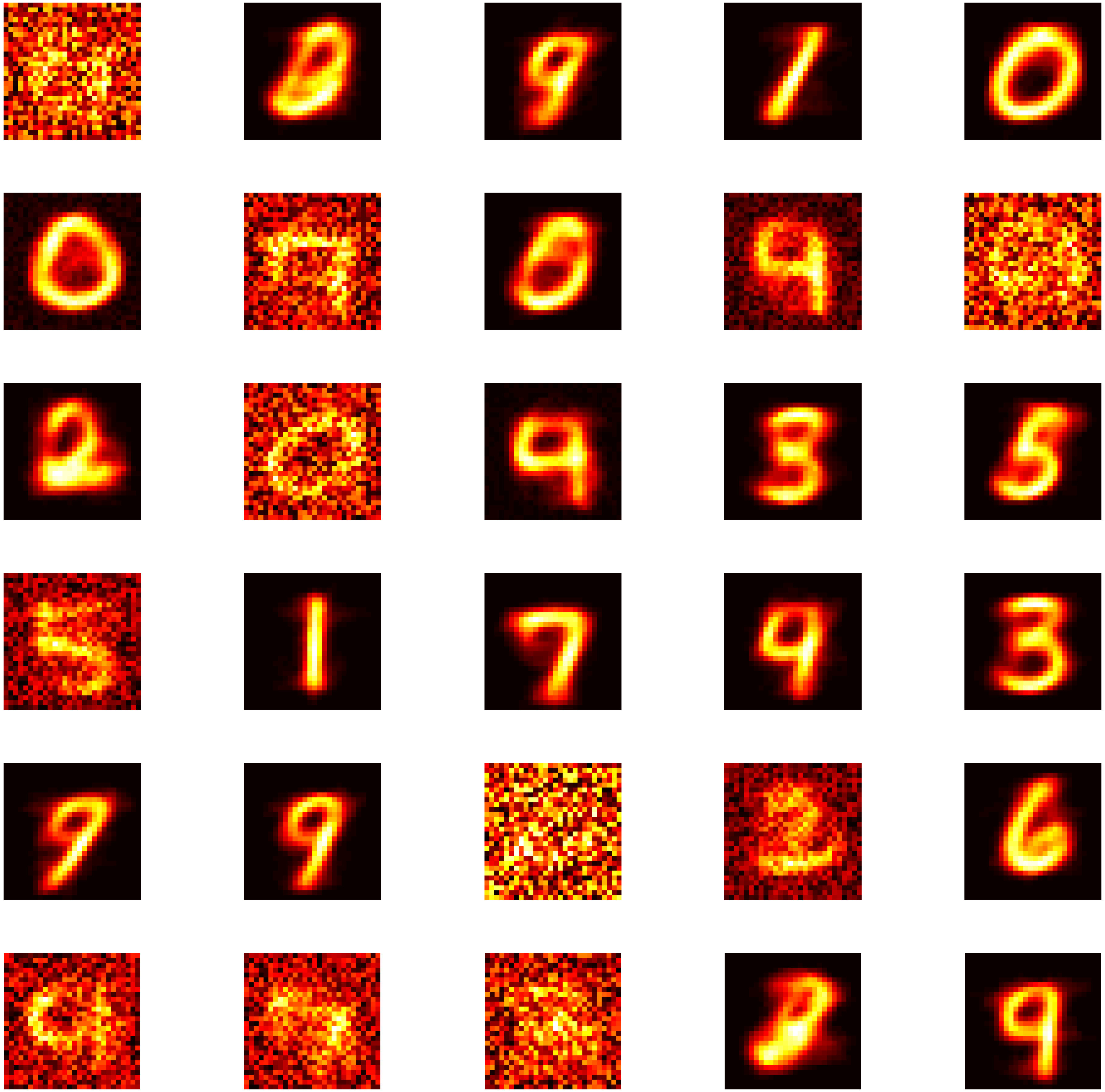


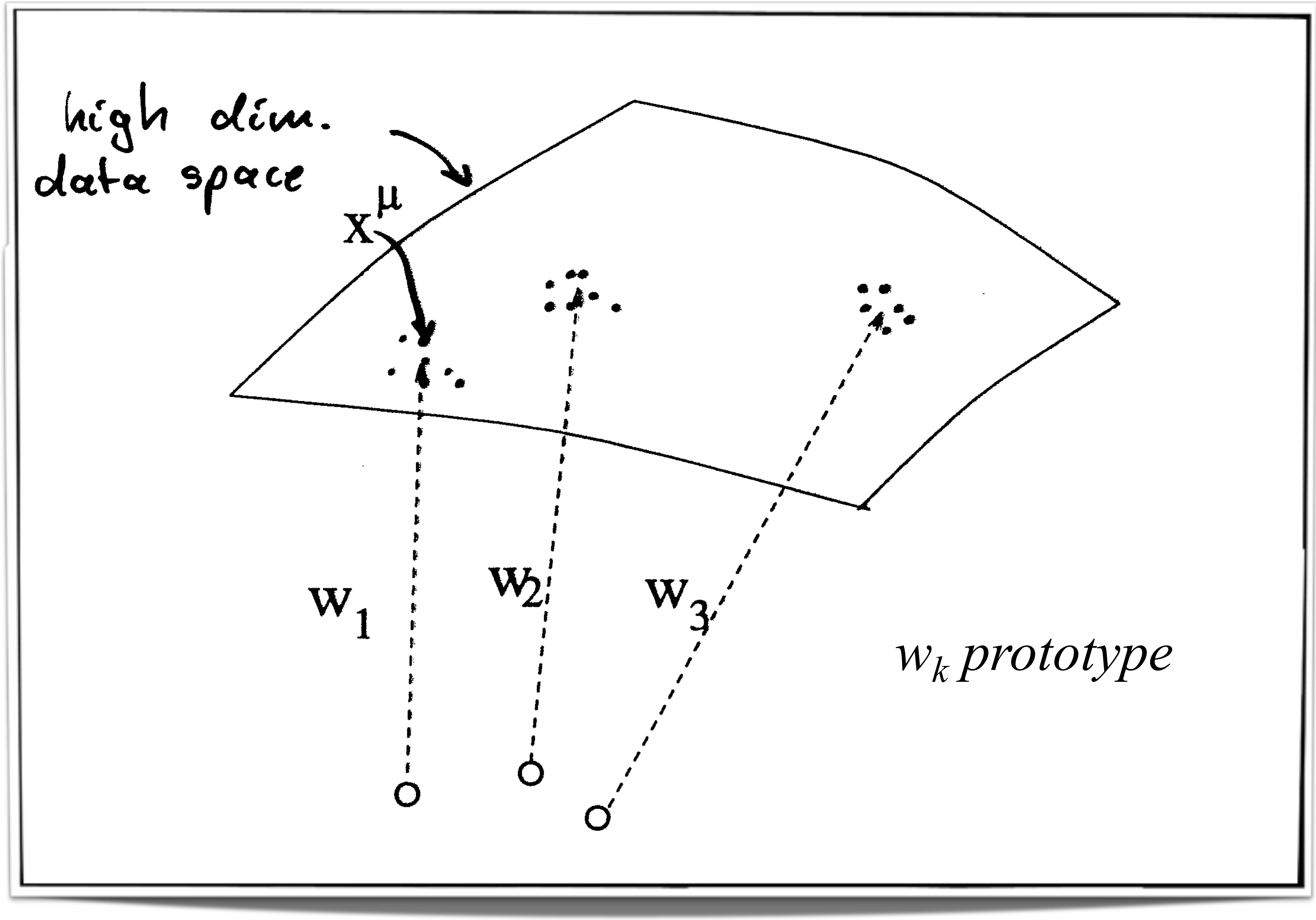
28x28=784

Network ?



Responce Level
0.02
0.015
0.01
0.005
0
5
10
15
20
25
30
Neuron index
5
10
15
20
25
5
10
15
20
25
5
10
15
20
25
5
10
15
20
25
Mean Weight Change
0.25
0.2
0.15
0.1
0.05
0
0
2000
4000
6000
8000
10000
12000
Time





$$\left|\mathbf{w}_k - \mathbf{x}^\mu\right| \leq \left|\mathbf{w}_i - \mathbf{x}^\mu\right| \quad \texttt{for all } i$$

Competition

Claim: statement (1) is equivalent to statement (2), where k is the index of the winning prototype (or neuron), assuming that weights are normalised.

(1) $$\mathbf{w}_k^T \mathbf{x}^\mu \geq \mathbf{w}_i^T \mathbf{x}^\mu \qquad \texttt{for all } i$$

(2) $$\left|\mathbf{w}_k - \mathbf{x}^\mu\right| \leq \left|\mathbf{w}_i - \mathbf{x}^\mu\right| \qquad \texttt{for all } i$$

Competition

Statement (1): assign pattern to the neuron that responds maximally.

Statement (2): assign pattern to the most similar prototype.

$$(1) \qquad \mathbf{w}_k^T \mathbf{x}^\mu \geq \mathbf{w}_i^T \mathbf{x}^\mu \qquad \texttt{for all } i$$

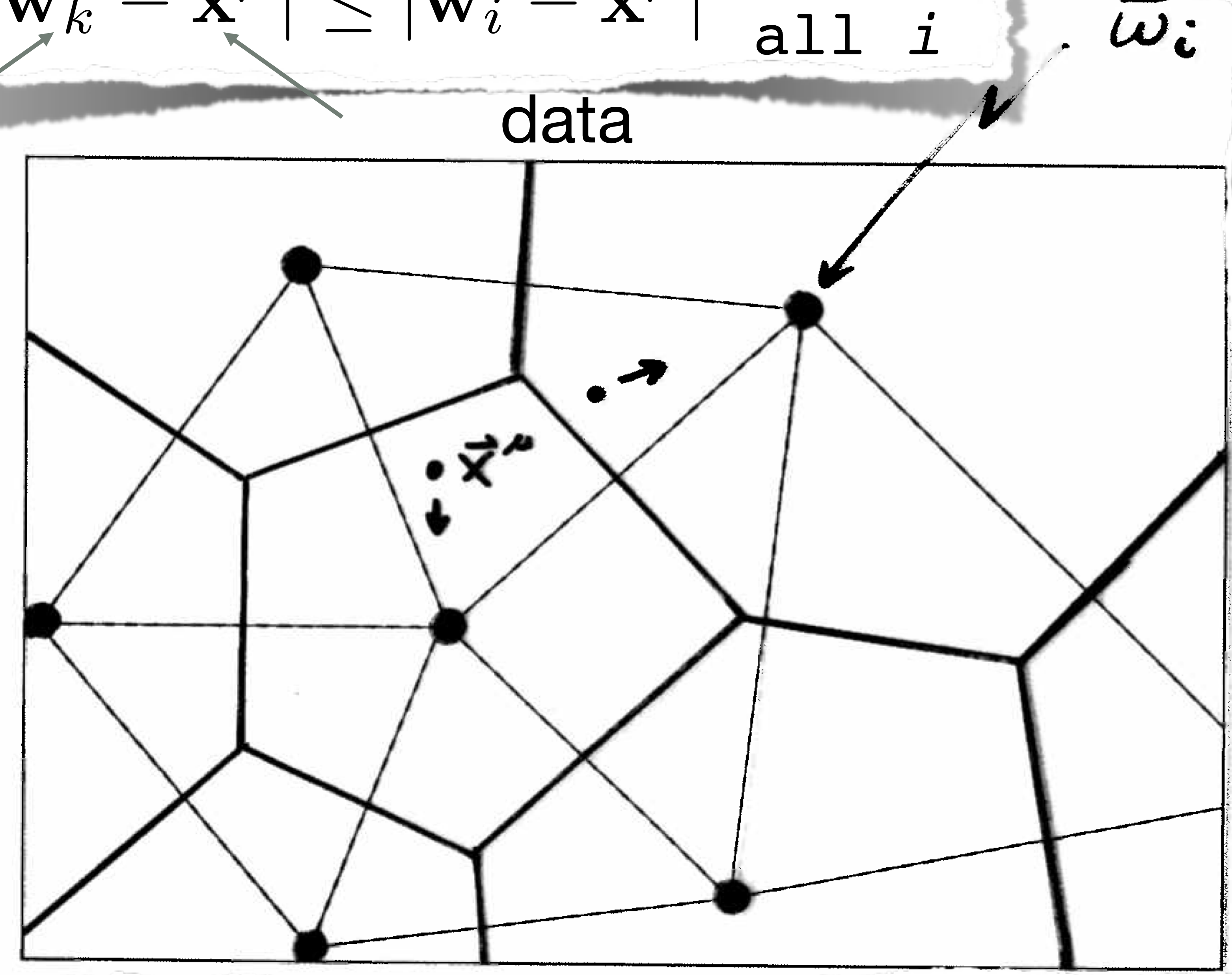
$$(2) \qquad |\mathbf{w}_k - \mathbf{x}^\mu| \leq |\mathbf{w}_i - \mathbf{x}^\mu| \qquad \texttt{for all } i$$

Competition

$$|\mathbf{w}_k - \mathbf{x}^\mu| \leq |\mathbf{w}_i - \mathbf{x}^\mu| \quad \texttt{for all } i$$

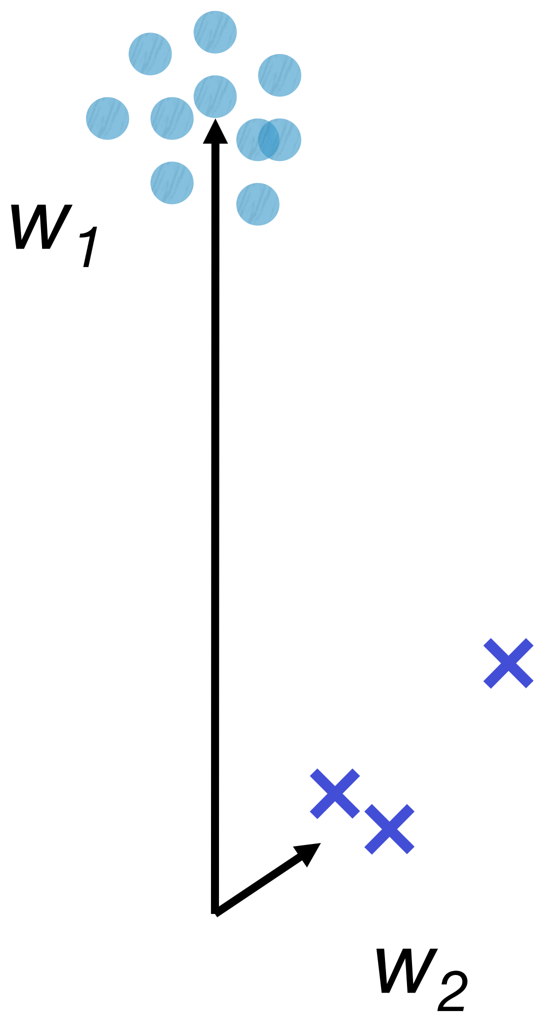
prototype

data



Voronoi (or Dirichlet) tesselation

Competition



$$\mathbf{w}_k^T \mathbf{x}^\mu \geq \mathbf{w}_i^T \mathbf{x}^\mu$$

Which prototype wins ?

Critical Point:
Normalisation Of Weights

K-means clustering

1) Assign $\mathbf{x}^{\mu} \rightarrow \mathbf{w}_k$ **(k is the nearest prototype)**

$$\left|\mathbf{w}_k - \mathbf{x}^{\mu}\right| \leq \left|\mathbf{w}_i - \mathbf{x}^{\mu}\right|$$ **for all i**

2) Update *only* the weights of the nearest prototype:

$$\Delta \mathbf{w}_k = \alpha\left(\mathbf{x}^{\mu} - \mathbf{w}_k\right)$$ **on-line rule**

Have you seen this rule before? $\Delta \mathbf{w} = \alpha\left(y\mathbf{x}^{\mu} - y^2\mathbf{w}\right)$

Competitive Learning

1) Assign $\mathbf{x}^{\mu} \rightarrow \mathbf{w}_k$ **(k is the winning neuron)**

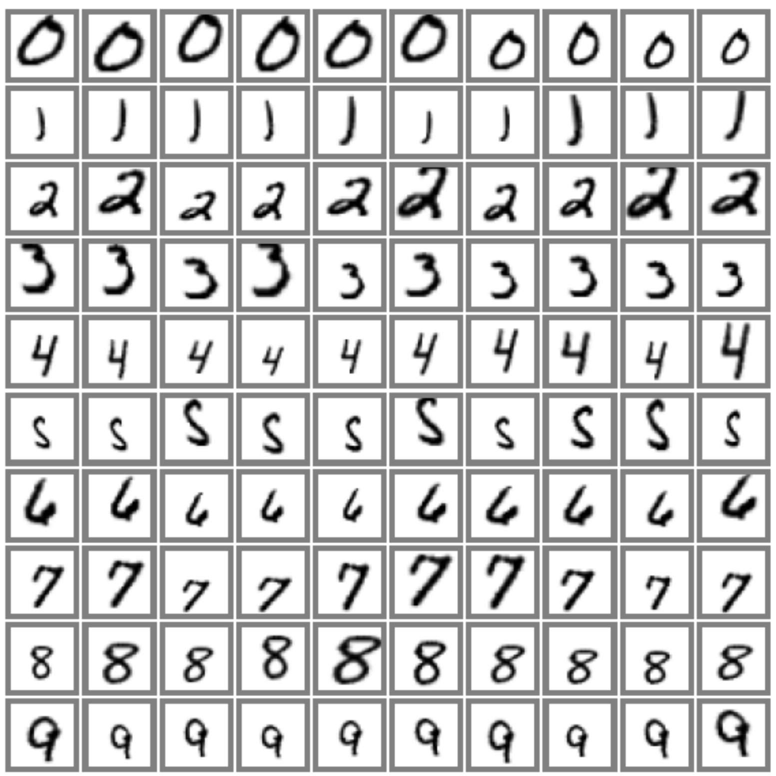
$$\mathbf{w}_k^T \mathbf{x}^{\mu} \geq \mathbf{w}_i^T \mathbf{x}^{\mu}$$ **for all i**

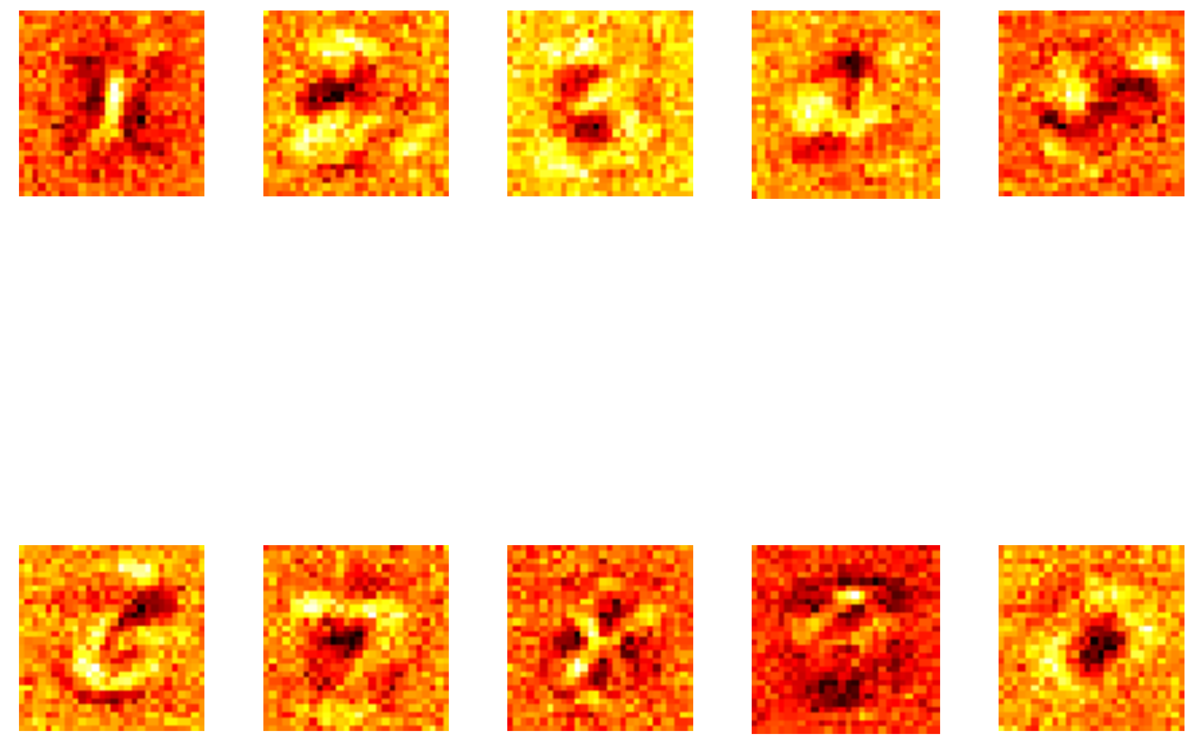
2) Update only the weights of the winning neuron by:

$$\Delta \mathbf{w}_k = \alpha(\mathbf{x}^{\mu} - \mathbf{w}_k)$$ **on-line rule**

Oja's rule $$\Delta \mathbf{w} = \alpha(y\mathbf{x}^{\mu} - y^2\mathbf{w})$$

Competitive Learning





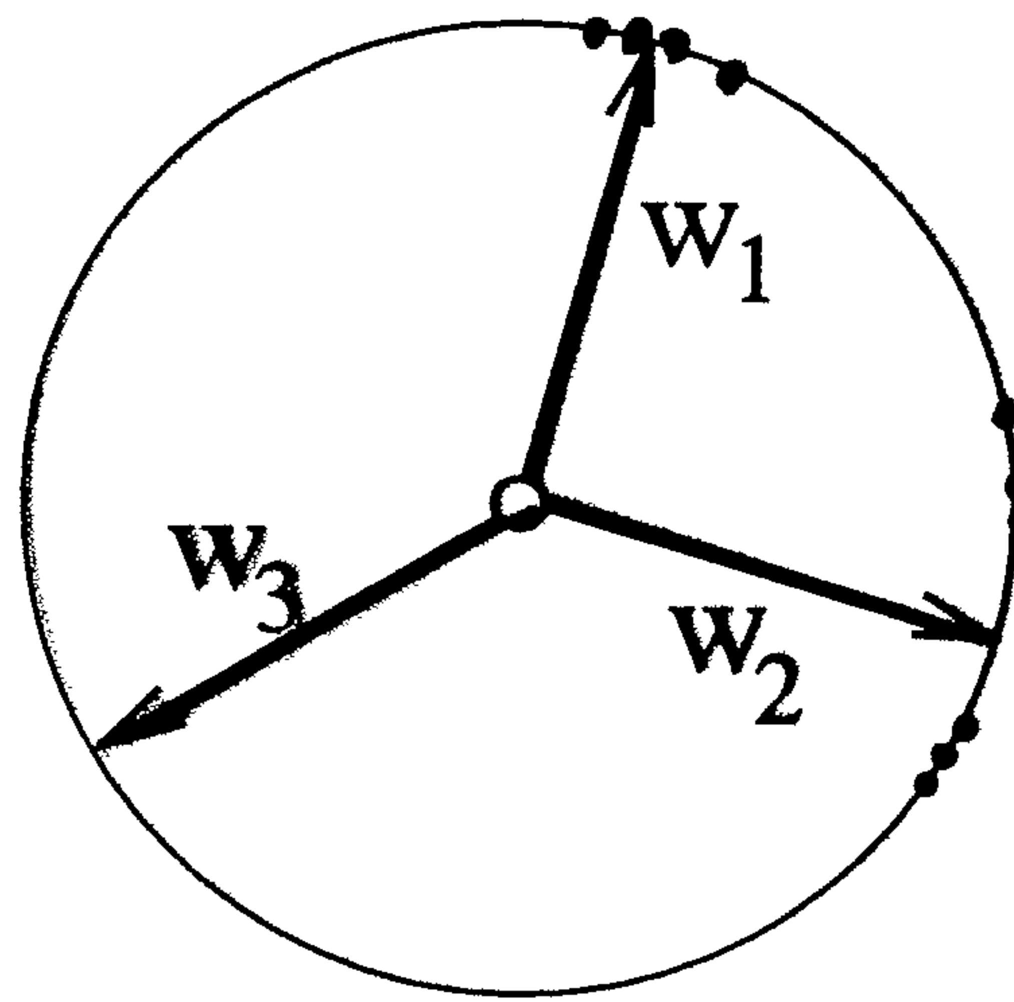
How weights should be adapted in order to look like the data?

$$\Delta \mathbf{w}_k = \alpha(\mathbf{x}^\mu - \mathbf{w}_k)$$

Question: why not $\Delta \mathbf{w}_k = \alpha \mathbf{x}^\mu$

The problem of dead units

- Initialise the weights to data samples
- Lucky learning: update winners & losers (losers with a smaller learning rate)
- Update the winners and neighbouring losers
- Add a bias that makes more difficult winning neurons to win again (but: stability issues)
- Add noise



w_3 corresponds to a dead unit: it has not learned any pattern

Summary

- Receptive field development: complex neurons can be formed by receiving appropriately wired input from simpler ones.
  - Wiring: hebbian learning.
- An idea that allows to maximise variability among neurons is competition.
  - K-means is the same as competitive learning (under specific conditions).
  - Oja's rule is the same as Competitive learning rule (under specific assumptions).

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