Learning in Neural Networks: Lecture 3

Competitive Learning with Hebbian rules

Wulfram Gerstner

EPFL, Lausanne, Switzerland

Introduction

Interacting neurons: weak linear interaction

Winner-take-all: strong inhibitory interaction

K-means clustering can be implemented by a Hebbian rule

Soft competition and representation learning

Development of Receptive fields

Previous slide.

The aim is to show that several clustering algorithm (e.g. K-means) but also a variant of Mixture Models can be implemented by Neural Networks that use Hebbian learning.

This is a classic topic of Neural Network theory and is covered in textbooks, such as the Book of Simon Haykin or the book of Hertz-Krogh-Palmer

Hebbian Learning (LTP)

Hebbian coactivation:
pre-post-post-post

"if two neurons are active together, the connection between those two neurons gets stronger."

"another synapse (red) which does not receive presynaptic spikes, does NOT increase"

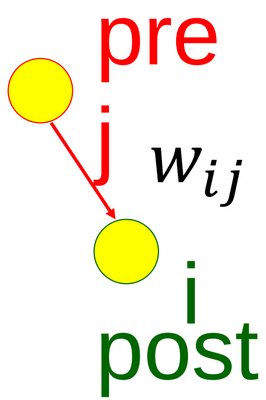
Previous slide.

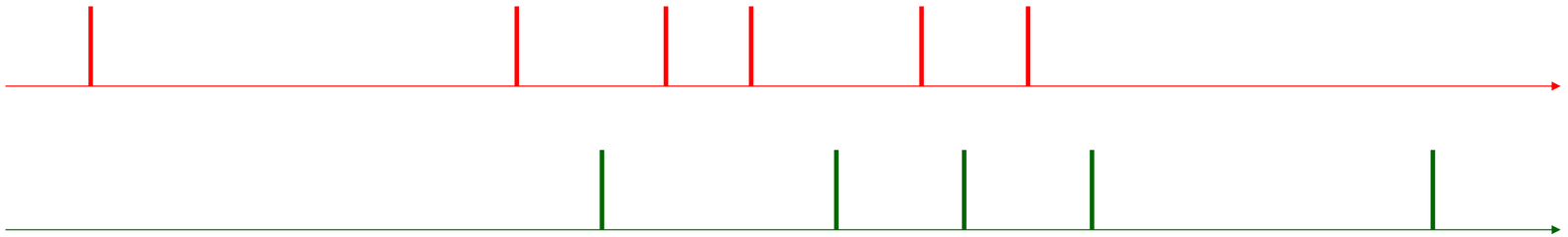
The joint activation of pre- and postsynaptic neuron induces a strengthening of the synapses. A strong stimulus is several repetitions of a pulse of the presynaptic neuron, followed by three or four spikes of the postsynaptic neuron.

Hundreds of experiments are consistent with Hebbian learning.

Note that by definition of Hebbian learning, only the stimulated synapses (green) is strengthened, but not another synapses (red) onto the same neuron.

Rate-based Hebbian Learning





Blackboard1

Local rule:

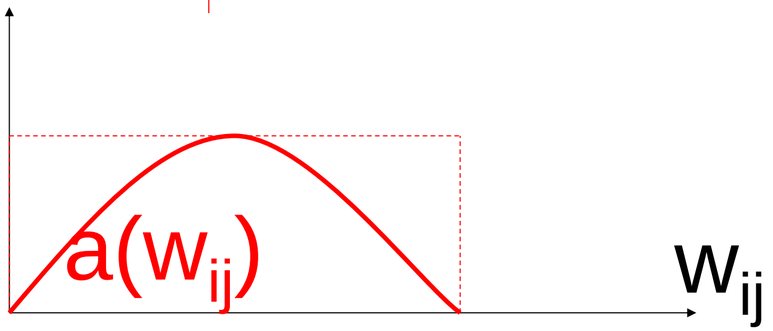
$$\Delta w_{ij} = F(w_{ij}, MOD; \nu_j^{pre}, \nu_i^{post})$$

Modulator MOD=const

Taylor expansion:

$$\Delta w_{ij} = a_0 + a_1^{pre}\nu_j^{pre} + a_1^{post}\nu_i^{post} + a_2^{corr}\nu_j^{pre}\nu_i^{post} + a_2^{post}\left(\nu_i^{post}\right)^2 + a_2^{pre}\left(\nu_i^{pre}\right)^2 \ldots$$

$a = a(w_{ij})$

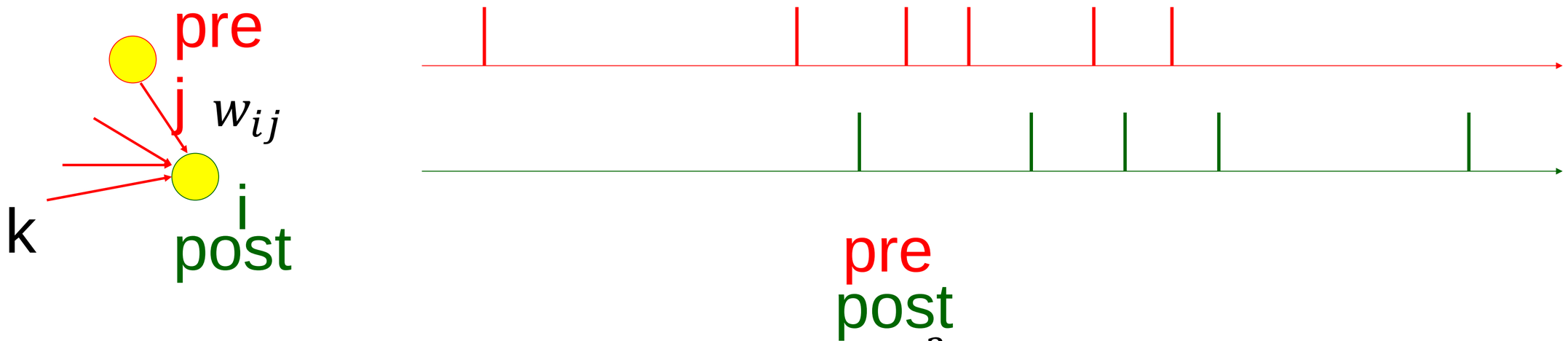


Previous slide.
Let us formulate these insights mathematically.

(i) Local rule implies that the weight change Δw_{ij} depends explicitly only on th firing rate of the pre- and postsynaptic neuron. It can also depend on the momentary value of the synaptic weight w_{ij} itself. Finally, it could also depend on other factors, for example on the presence or absence of a neuromodulator such as dopamine, called *MOD*. At the moment we assume that the value of *MOD* does not change so that we can disregard it.

(ii) The Hebbian rule says little about the function F. We assume that F allows a Taylor expansion. We expand F with respect to the two firing rates, but not with respect to the weight value itself. As a result we have expansion coefficients that still carry the weight-dependence as an argument.

2. Rate-based Hebbian Learning



$$\Delta w_{ij} = a_2^{corr} \nu_j^{pre} \nu_i^{post} - w_{ij} \left(\nu_i^{post}\right)^2$$ Oja-rule (for linear neurons)

$$\Delta w_{ij} = a_2^{corr} \nu_j^{pre} \nu_i^{post} \left(\nu_i^{post} - \vartheta\right)$$ BCM-rule

$$\Delta w_{ij} = a_2^{corr} (\nu_j^{pre} - \vartheta)(\nu_i^{post} - \vartheta)$$ covariance-rule

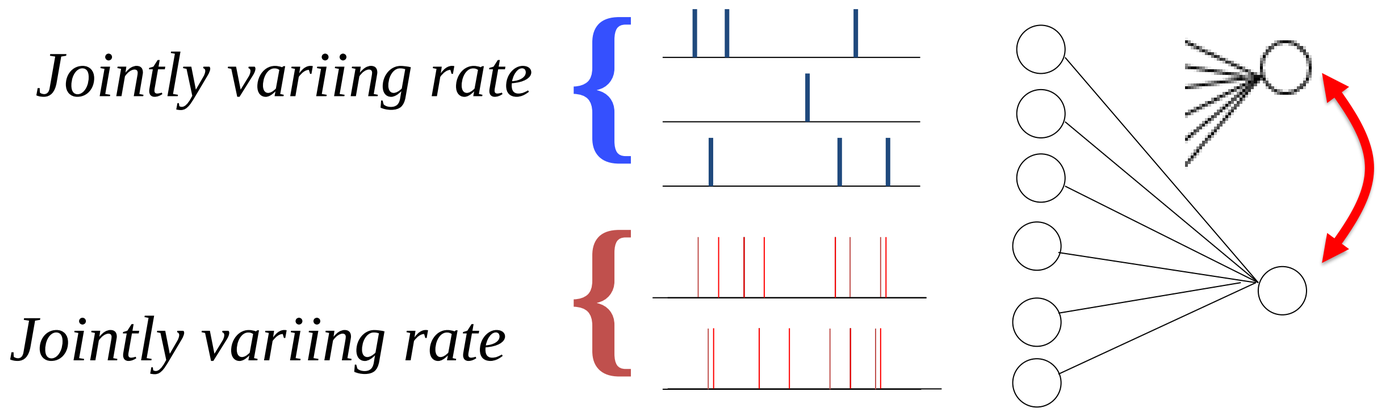
$$\Delta w_{ij} = a_2^{corr} \nu_j^{pre} \nu_i^{post} - w_{ij} \left(\nu_i^{post}\right)$$ competitive-rule (for non-linear neurons)

Previous slide.

The first three Hebbian rules are examples that we have seen previously.

The competitive rule at the end is the one we use later today.

Functional Consequence of Hebbian Learning



Hebbian Learning detects correlations in the input

So far: considered a single neuron at a time.

Now: interacting neurons!

Weak inhibitory interaction → neurons decorrelate.

Strong inhibitory interaction → neurons compete → winner.

Previous slide.
We saw that Hebbian learning can focus on correlated input, but so far we mostly considered one single neuron at a time.

The question now is: how can we ensure that different neurons focus on different aspects of the input?

The answer is by applying inhibitory connections between neurons.

Quiz: biological neural networks

[ ] Neurons are nonlinear

[ ] The total input to a neuron is the weighted sum of individual inputs

[ ] The neuronal network in the brain is feedforward: it has no lateral connections

[ ] Hebb-rules are always linear in 'ν_j^{pre}' and 'ν_i^{post}'

[ ] 2-factor rules can always be written as a multiplication of a 'pre'-term with a 'post'-term

Previous slide.

Learning in Neural Networks: Lecture 3

Competitive Learning with Hebbian rules

Wulfram Gerstner

EPFL, Lausanne, Switzerland

Introduction

Interacting neurons: weak linear interaction

Winner-take-all

K-means clustering

Soft competition

P. Foldiak (1989), Adaptive network for optimal linear feature extraction, IEEE https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=118615&tag=1

Gerstner and Brito (2016), Nonlinear Hebbian learning as a unifying principle, *PLOS Comput. Biol. .*

Vogels et al. (2011), Inhibitory plasticity balances excitation andinhibition. *Science*

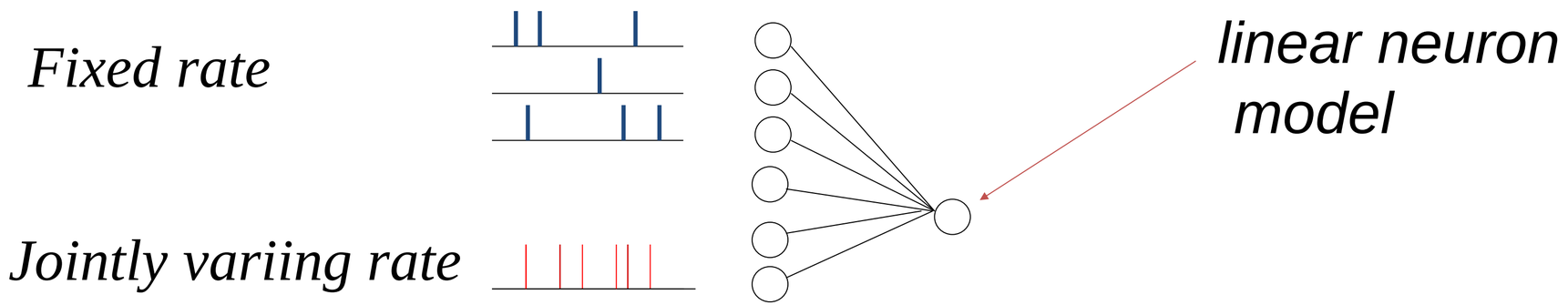
Previous slide.
We saw that Hebbian learning can focus on correlated input, but so far we mostly considered one single neuron at a time.

The question now is: how can we ensure that different neurons focus on different aspects of the input?

The answer is by applying inhibitory connections between neurons.

,

Summary from Lecture 1: Hebbian Learning for PCA



- **Hebbian learning detects correlations in the input**
- **linear neuron model and Oja plasticity rule**

$$\frac{d}{dt}w_{ij} = a_2^{corr} v_i^{post} v_j^{pre} - w_{ij}(v_i^{post})^2$$

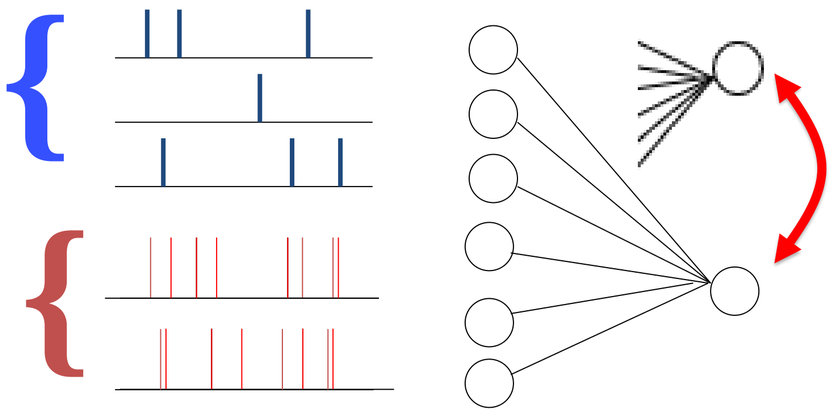
- **Hebbian learning aligns weight vector with first PC of correlation matrix**
- **BUT: only one PC extracted**

Previous slide.
PC = Principal component = eigenvector of correlation matrix with maximum eigenvalue

Add interactions: Hebbian Learning for multiple PCs

Jointly variing rate

Jointly variing rate



$$v_i^{post} = \sum_k w_{ik}\, v_k^{pre} - \sum_n B_{in} v_n^{post}$$

inhibitory

Hebbian Learning detects multiple PCs

Weak inhibitory interaction → neurons decorrelate.

Hebb/Oja $\frac{d}{dt} w_{ij} = a_2^{corr} v_i^{post} v_j^{pre} - w_{ij} (v_i^{post})^2$

Hebb $\frac{d}{dt} B_{in} = +a^{lat} v_i^{post} v_n^{post}$

neurons extract different PCs
→ converges to PCA subspace

P. Foldiak, Adaptive network for optimal linear feature extraction, 1989

Previous slide.
We can add linear interactions between the (linear) neurons.
If these interactions are negative (inhibitory), one neuron tells the others that they should focus on different inputs.

Moreover, one can make the lateral inhibitory weights learn with an Hebbian learning rule. Then the lateral weights grows to the subset of PCs that have the set of maximum eigenvalues of the correlation matrix. Say, PC 1,2,. … 5

As in lecture 1 we suppose that the data has mean zero. To understand how the interaction works imaging that

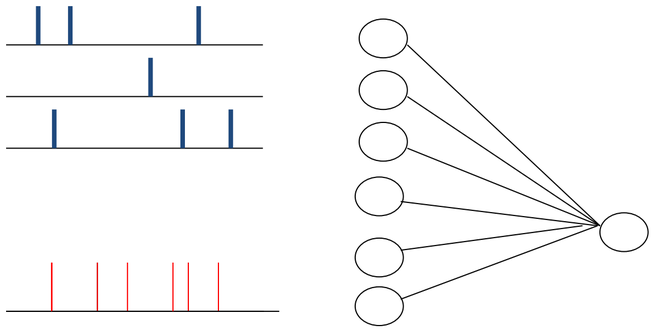
(i) Two output neurons are correlated. Then the lateral inhibition grows which makes the neurons decorrelated.
(ii) Two output neurons are anti-correlated (i.e. if one increases, the other one decreases. Then the interactions first decrease and then change sign so that correlation increases - which makes them decorrelated.

As a result of these interactions the input weight vectors become orthogonal, so that the output neurons are decorrelated and the lateral interactions are close to zero.

Summary from Lecture 2: Hebbian Learning for ICA

input data:

- *Centered at zero*
- *Whitened*



nonlinear neuron model

- **Hebbian learning detects independent components in the input**
- **non-linear neuron model and simple plasticity rule**

$$\frac{d}{dt} w_{ij} = a_2^{corr} \nu_i^{post} \nu_j^{pre} = a_2^{corr} g_i(\sum_k w_{ik}\, x_k) \nu_j^{pre}$$ + renormalize $\vec{w} = \frac{\vec{w}^{new}}{|\vec{w}^{new}|}$

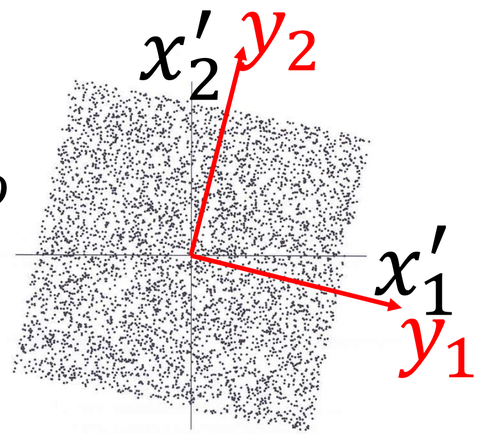
- **Hebbian learning aligns weight vector with first IC of data**
- **BUT: only one IC extracted**

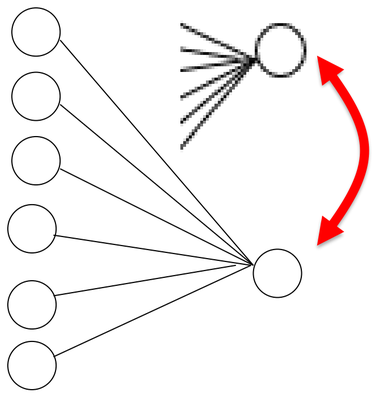
Previous slide.

IC = independent component = direction of maximal or minimal non-Gaussianity of data distribution

Add interactions: Hebbian Learning for multiple ICs

input data:
- *mean zero*
- *whitened*





$$v_i^{post} = g(\sum_k w_{ik}\, v_k^{pre}) - \sum_n B_{in} v_n^{post}$$

inhibitory

Hebbian Learning detects orthogonal ICs

Weak inhibitory interaction → output neurons decorrelate.

Hebb $\frac{d}{dt} w_{ij} = a_2^{corr} v_i^{post} v_j^{pre}$ + renormalize $\vec{w} = \frac{\vec{w}^{new}}{|\vec{w}^{new}|}$

Hebb $\frac{d}{dt} B_{in} = +a^{lat}(v_i^{post} - \overline{v_i^{post}}) v_n^{post}$ Neurons become decorrelated

Gerstner and Brito (2016), Nonlinear Hebbian learning as a unifying principle, *PLOS Comput. Biol.* .

Vogels et al. (2011), Inhibitory plasticity balances excitation andinhibition. *Science*

Previous slide.
We add linear interactions between the (nonlinear) neurons.
If these interactions are negative (inhibitory), one neuron tells the others that they should focus on different inputs. The lateral weights learn with an Hebbian learning rule. Then the forward weight vectors grow to a set of orthogonal ICs.

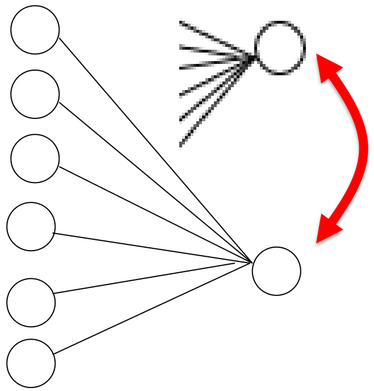
As in lecture 2 we suppose that the data has mean zero and is whitened. The argument is as before. Imagine that that

(i) Two output neurons are correlated. This means that the input weight vectors are not orthogonal but have an angle below 90 degree. Then the inhibition grows which decreases correlation by turning the input weight vector. The growth stops at 90 degree.

(ii) Two output neurons are anti-correlated. This is only possible if one of the output signals driving learning can turn negative. Therefore subtract a running average ν_i^{post}. Then angles of input weight vectors larger than 90 degree lead to a reduction of the lateral interaction.

As a result of these interactions the input weight vectors become orthogonal, so that the output neurons are decorrelated.

Quiz:



$$v_i^{post} = g(\sum_k w_{ik}\, v_k^{pre}) - \sum_n B_{in} v_n^{post}$$

To extract multiple Independent Components (ICs)

[ ] the forward weights can be learned with a standard Hebb rule ‘pre times post’

[ ] there is no need to whiten the data

[ ] A Hebbian rule ‘pre times post’ for inhibitory later weights (red) works well to ensure that different neurons extract different ICs

[ ] After long enough training, different ICs always have nearly always weight vectors that have an angle close to 45 degree to each other

Previous slide.

Learning in Neural Networks: Lecture 3

Competitive Learning with Hebbian rules

Wulfram Gerstner

EPFL, Lausanne, Switzerland

Introduction

Interacting neurons: weak linear interaction

Interacting neurons: Winner-take-all

K-means clustering

Soft competition

P. Foldiak (1989), Adaptive network for optimal linear feature extraction, IEEE https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=118615&tag=1

Gerstner and Brito (2016), Nonlinear Hebbian learning as a unifying principle, *PLOS Comput. Biol. .*

Vogels et al. (2011), Inhibitory plasticity balances excitation andinhibition. *Science*

Previous slide.

So far we assumed relatively weak lateral interactions.

Now we assume that inhibitory interactions are strong so that if one neuron is active it shuts down all other neurons.

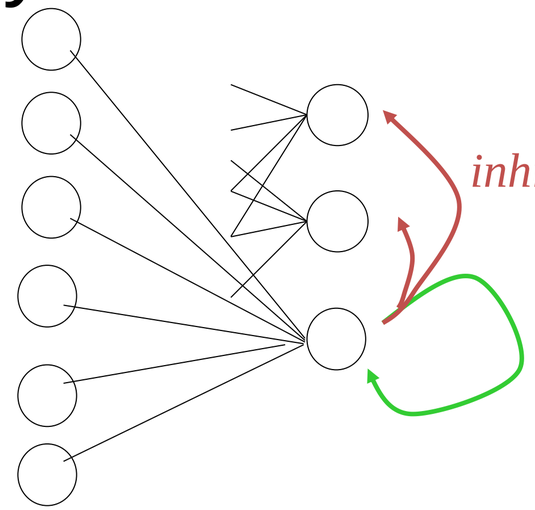
Winner-take-all circuit: strong lateral inhibition

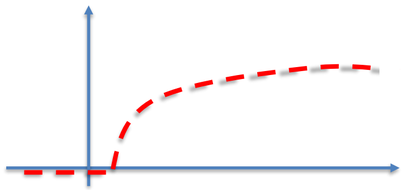
- Nonlinear neurons with positive rate $\nu_i^{post} \geq 0$
- Strong inhibitory interactions $\nu_i^{post} = g(\sum_k w_{ik}\, \nu_k^{pre} - \sum_n B_{in} \nu_n^{post})$

- Activity of one neuron suppresses activity of all others

$\nu_j^{pre} = x_j^{\mu}$

input pattern $\vec{x}^{\mu}$





Loop

1) initialize $\nu_i^{post} = 0$
2) apply input $\vec{x}^{\mu}$
3) **Loop**: run dynamics (*) for **several** steps Δt

➡ update dynamics? - discrete time steps

$(*)\ \nu_i^{post}(t + \Delta t) = g\left(\sum_k w_{ik}\, \nu_k^{pre}\ (t) - \sum_n B_{in} \nu_n^{post}(t)\right)$

Blackboard

Previous slide.

1. We assume that the nonlinear function g is a saturating function which vanishes below some threshold. The maximum output is 1.

2. We add **strong** inhibitory interactions $B_{in} > 1$ between the (nonlinear) neurons. Note that inhibition enters into the argument of the non-linearity.

3. We also add positive self-interactions of each neuron with each self. This can lead to an instability that rapidly drives the activity of a neuron to its maximum.

4. However, the neurons that reaches the maximum first, inhibits the others so strongly that it is impossible that more than one neuron is active. The active neuron is called the winner.

5. The dynamics is updated iteratively **with fixed input**:

$$\nu_i^{post}(t + \Delta t) = g(\sum_k w_{ik}\, \nu_k^{pre}\ (t) - \sum_n B_{in} \nu_n^{post}\ (t))$$

6. The input consists of patterns $\vec{x}^{\mu}$ so that $\nu_j^{pre} = x_j^{\mu}$ for several time steps

Hebbian learning rule

Neuronal dynamics rapidly converges to 'winner':

$$\nu_i^{post} = \delta_{ik}$$

-k is winner
-Only k is active

➡ How can we update the weights ?

$$\Delta w_{ij} = \eta \left(\nu_i^{post} x_j^{\mu} - \gamma w_{ij} \nu_i^{post} \right)$$

-k is winner
-Only k is active

$$\Delta \vec{w}_k = \eta (\vec{x}^{\mu} - \vec{w}_k)$$

→ on-line update (one input at a time)

→ updates only for winner (one neuron at a time)

Previous slide.

1. The input consists of patterns $\vec{x}^{\mu}$, hence $\nu_j^{pre} = x_j^{\mu}$

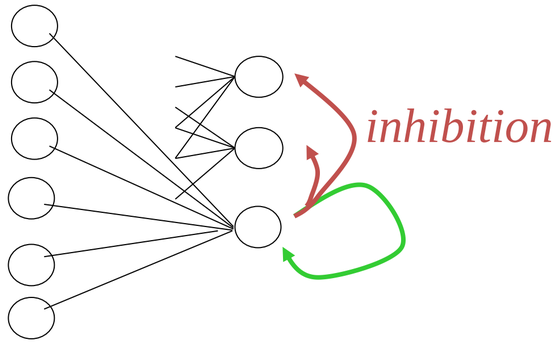
2. After convergence of the network dynamics, a single neuron k is winner. The winner has activity equal to 1, all other neurons have zero activity.

3. We apply Hebbian learning to ALL output neurons of the network. However, with the proposed learning rule, only the winner adapts its weights because only the winner has non-zero firing rate. The weights of the other neurons remain unchanged.

4. The weight vector of the winner moves towards the current pattern $\vec{x}^{\mu}$ (see next slide)

Winner-take-all circuit: strong lateral inhibition

input pattern $\vec{x}^{\mu}$

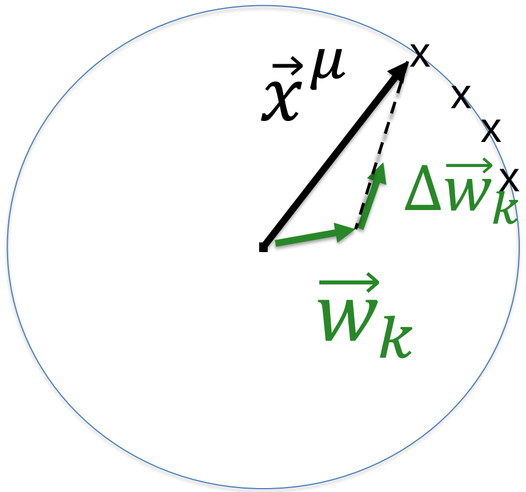


for winner:

$$\Delta\vec{w}_k = \eta(\vec{x}^{\mu} - \vec{w}_k)$$

Suppose all input pattern $\vec{x}^{\mu}$ have $|\vec{x}^{\mu}|^2 = 1$ and $0 < v_j^{pre} = x_j^{\mu}$

Then $\rightarrow |\vec{w}|^2 \approx 1$

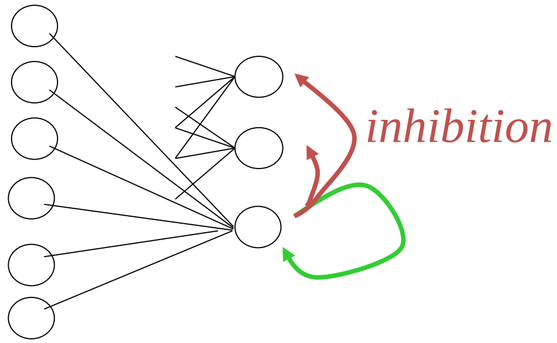


Previous slide.
Because of the normalization, all data points lie on the sphere.

The weight vector moves towards the data point. By assumption data lies on the positive quadrant of the sphere. Therefore the weight vector will lie inside, but has its end point close to the surface of the sphere.

Winner-take-all circuit: strong lateral inhibition

input pattern $\vec{x}^{\mu}$



several output neurons k, with input weight vectors $\vec{w}_k$

Suppose all input pattern $\vec{x}^{\mu}$ have $|\vec{x}^{\mu}|^2 = 1$ and $|\vec{w}_k|^2 = 1$.

Then:

→ $\min_k\{|\vec{x}^{\mu} - \vec{w}_k|\}$ **equivalent to** $\max_k\{\vec{w}_k^T \vec{x}^{\mu}\}$

Blackboard 2

Previous slide.

One line calculation. (blackboard)

We assume that for data point $\vec{x}^{\mu}$ the weight vector $\vec{w}_k$ with index k is the closest one

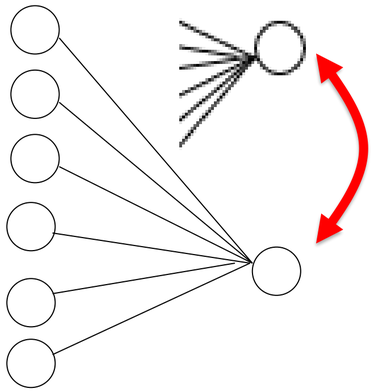
$$(1) \qquad |\vec{w}_k - \vec{x}^{\mu}| \leq |\vec{w}_i - \vec{x}^{\mu}| \qquad \text{for all } i$$

We square the equation (1) and multiply out all terms. Then with our assumptions $|\vec{x}^{\mu}|^2 = 1$ and $|\vec{w}_k|^2 = 1$ the statement

$$(2) \qquad \vec{w}_k^{\mathrm{T}} \vec{x}^{\mu} \geq \vec{w}_i^{\mathrm{T}} \vec{x}^{\mu} \qquad \text{for all } i$$

follows.

Quiz:



$$v_i^{post} = g(\sum_k w_{ik}\, v_k^{pre} - \sum_n B_{in} v_n^{post})$$

[ ] To select a single winner via lateral interactions, lateral interactions should be inhibitory

[ ] For updating the weights, the learning rule can be applied to all neurons

[ ] Only the winning neuron updates its forward weights

[ ] a positive self-interaction loop is helpful, but not absolutely necessary

Feedback on competition by lateral interaction

[ ] At least 60 percent of the material on competive networks was new to me

For 80 percent of the material that we have seen in this part
[ ] I understood the concepts and got a good idea of the formalism

Learning in Neural Networks: Lecture 3

Competitive Learning with Hebbian rules

Wulfram Gerstner

EPFL, Lausanne, Switzerland

Introduction

Interacting neurons: weak linear interaction

Interacting neurons: Winner-take-all

K-means clustering

Grossberg (1976) Adaptive Pattern Classification and Universal Recoding:
I. Parallel Development and Coding of Neural Feature Detectors, *Biol, Cybern.*
Rumelhard and Zipser (1985) Feature discovery by competitive learning, Cognitive Science
Hertz-Krogh-Palmer (1991) Introduction to the theory of neural networks *(Addison-Wesley)*
Simon Haykin (1999) Neural Networks*, 2nd edition (Prentice Hall)*

Previous slide.

Now we connect the network dynamics and learning rule to a well-known algorithm: k-means clustering.

Most students have seen this algorithm already in other classes.

k-means clustering: "online" version has neuronal interpretation

Assume P data points $\vec{x}^{\mu}$ with $1 \leq \mu \leq P$

1) Pick a $\vec{x}^{\mu}$ and find the closest prototype

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

NETWORK DYNAMICS can do this

2) Change the weights of **this** prototypes by:

$$\Delta\vec{w}_k = -\eta(\vec{w}_k - \vec{x}^{\mu})$$

HEBBIAN LEARNING Can do this

3) Reduce η and go back to 1

'developmental Change'

Previous slide.
The classic k-means algo in its online version (one data point at a time).

This algorithm is found in many introductory textbooks of signal processing and data science.

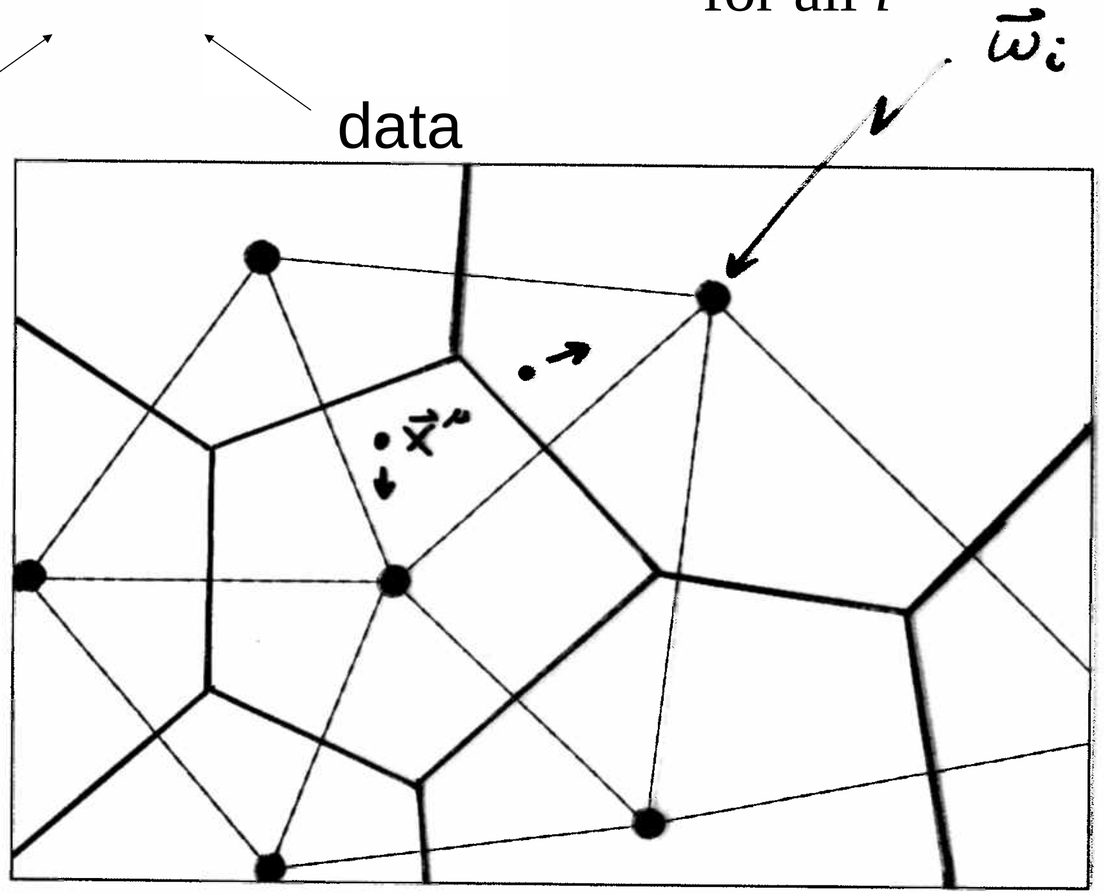
Based on the results of the previous sections, we now know that this algorithm can be implemented by Hebbian learning and lateral neuronal inhibition.

Classification by nearest prototype

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

prototype

data



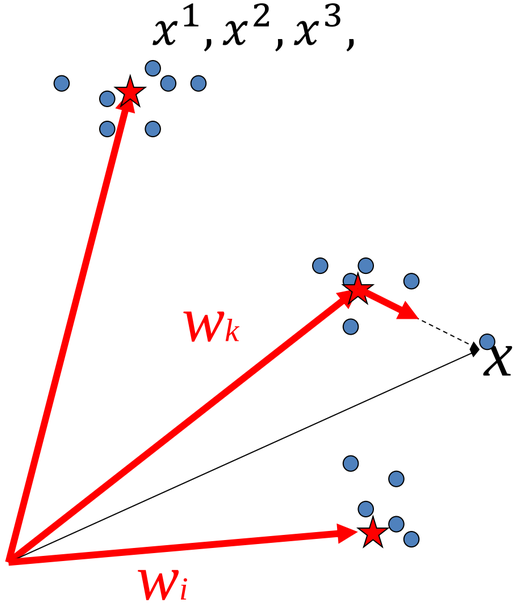
Voronoi (or Dirichlet) tesselation

Previous slide.

In the space of data points (here a plane), the classification by nearest prototype (nearest weight vector) induces a tessellation of the space.

k-means cClustering

Initialize: Prototypes w_k



Take a Datapoint x

1) Determine winner k

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

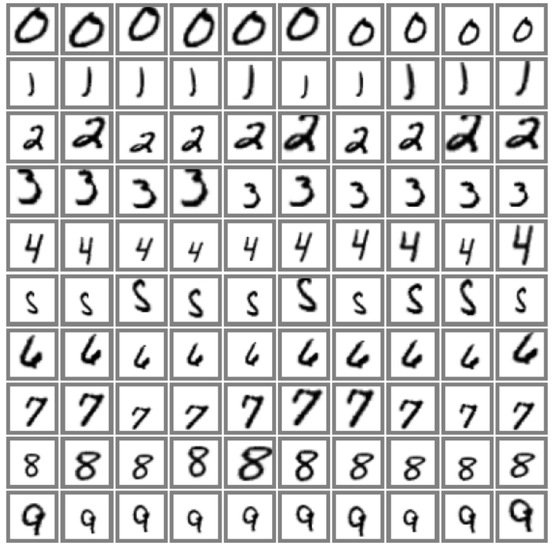
2) Update winner

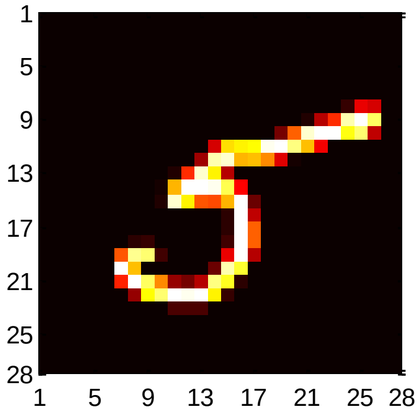
$$\Delta\vec{w}_k = \eta(\vec{x}^\mu - \vec{w}_k)$$

Moves the prototypes in direction of data points!

Previous slide.
A geometric illustration of the algorithm

Example: MNIST data, clustering of digits with Competitive Neural Network



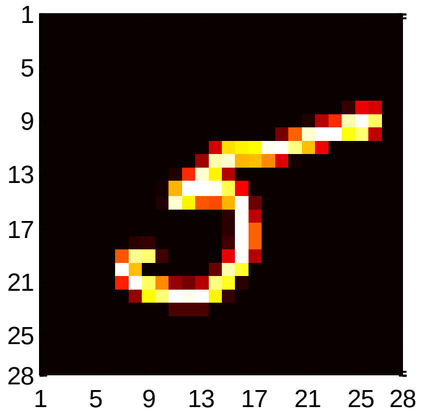


28x28=784

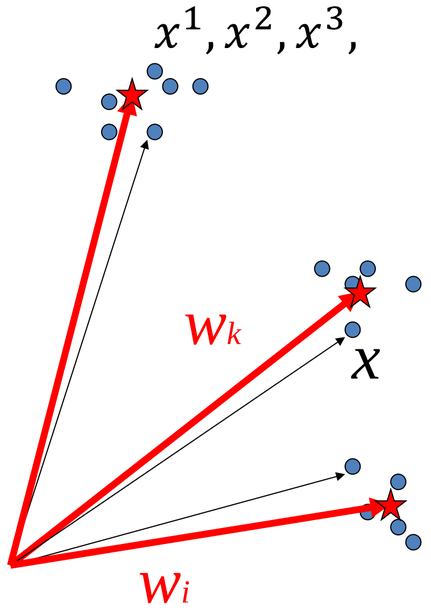
Previous slide.

Just a reminder of how the MINST data looks like.
Different writers have written several times the digits from 1 to 10. There are different writing styles. The digits are black on white background and (nearly) centered.

Hard Clustering/K-means clustering



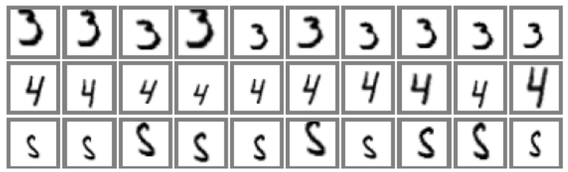
28x28=784



Group of similar points: ***cluster***

Prototype w_k 'represents group of data points'

Datapoint x



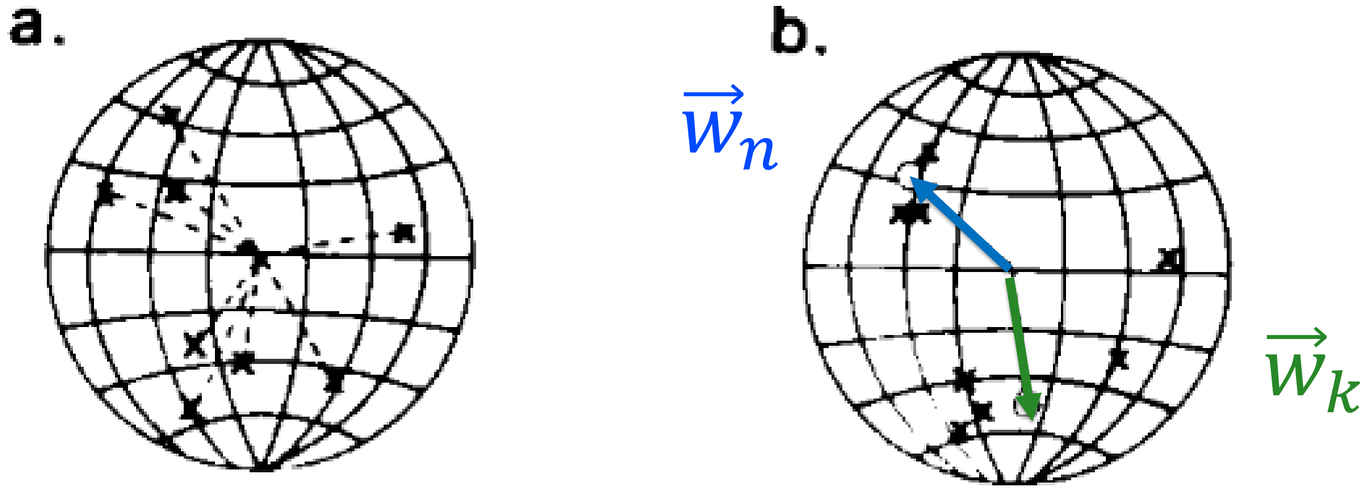
k winner: $|\vec{w}_k - \vec{x}^\mu| \leq |\vec{w}_i - \vec{x}^\mu|$ for all i

Previous slide.
Each sample of MNIST data is a point in the 784-dimensional space.
Images with similar configuration of images lie close to each other.

K-means clustering will move the weight vector (prototype) in the center of mass of the cluster.

Note that images that look similar to the human eye but where the handwritten digit is shifted by 3 pixels upwards or downwards look very different in pixel space!

Clustering of data on a sphere



weight vector moves to the center of mass of ‘its’ cluster

$$\Delta\vec{w}_k = -\eta(\vec{w}_k - \vec{x}^{\mu})\ \delta_{kj^*(\mu)}$$ online

$$\Delta\vec{w}_k = -\eta \sum_{\mu} \delta_{kj^*(\mu)}\ (\vec{w}_k - \vec{x}^{\mu})$$ batch

$j^*(\mu)$ =index of winner

Rumelhard and Zipser (1985) Feature discovery by competitive learning, Cognitive Science

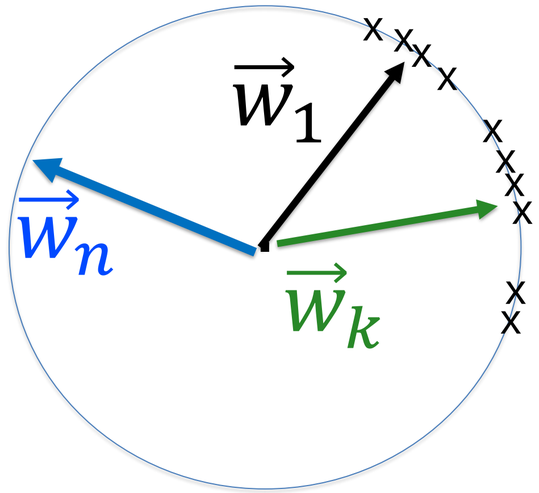
Previous slide.
The neural algorithm works best if the data points are represented by vectors of unit length. In this case all data points lie on the sphere.

A general result of k-means clustering is that the prototype (weight vector) moves towards the center of ‘its’ cluster of data points.

This implies that each weight vector is also (nearly) normalized to unit length. As a result, nearest neighbor is (nearly) equivalent to max scalar product.

See also exercises.

Dead unit



1) weight vector moves to the center of mass of ‘its’ cluster
2) A weight vector only moves if it the ‘winner’

→ Dead units possible

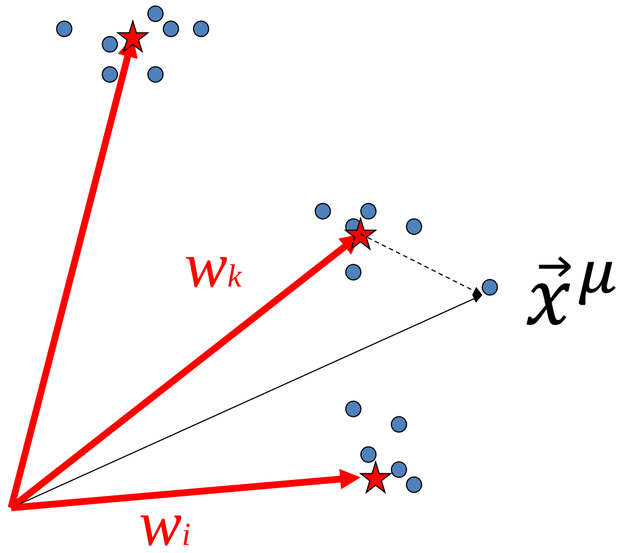
Dead units avoided by initializing weight vectors with ramdonly chosen data points

Previous slide.

The picture also shows that it can happen that some weight vector (here the blue vector) is responsible for no point in the data cloud. It is called a dead unit.

Theorem: Minimal Reconstruction Error

Suppose each data point is replaced by its closest prototype. Then k-means clustering minimizes the mean quadratic reconstruction error



reconstruction error

Sum over all data points

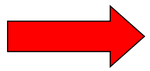
$$E = \sum_{k} \sum_{\mu \in C_k} (\vec{w}_k - \vec{x}^{\mu})^2$$

Cluster (prototype)

Samples in cluster k

Exercise: Derivation of the learning rule for k-means

$$E = \sum_{k} \sum_{\mu \in C_k} (\vec{w}_k - \vec{x}^{\mu})^2$$



Gradient descent on the error surface

$$\Delta \vec{w}_k = -\eta \sum_{\mu \in C_k} (\vec{w}_k - \vec{x}^{\mu})$$

batch

$$\Delta \vec{w}_k = \eta (\vec{x}^{\mu} - \vec{w}_k)$$

on-line

Feedback on k-means clustering

[ ] At least 60 percent of the material on k-mans clustering

For 80 percent of the material that we have seen in this part
[ ] I understood the concepts and got a good idea of the formalism

Previous slide.

Learning in Neural Networks: Lecture 3

Competitive Learning with Hebbian rules

Wulfram Gerstner

EPFL, Lausanne, Switzerland

Introduction

Interacting neurons: weak linear interaction

Interacting neurons: Winner-take-all

K-means clustering

Soft competition and Soft clustering

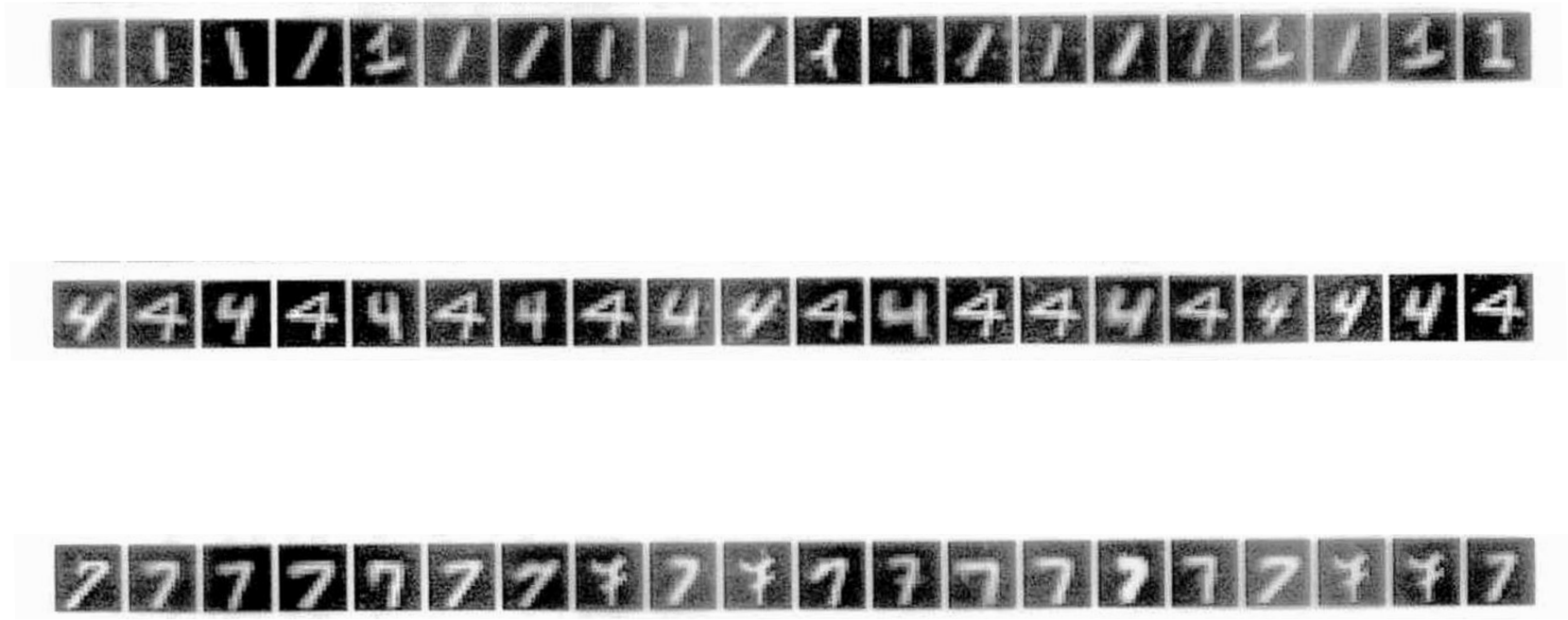
Grossberg (1976) Adaptive Pattern Classification and Universal Recoding:
I. Parallel Development and Coding of Neural Feature Detectors, *Biol, Cybern.*
Rumelhard and Zipser (1985) Feature discovery by competitive learning, Cognitive Science
Hertz-Krogh-Palmer (1991) Introduction to the theory of neural networks *(Addison-Wesley)*
Simon Haykin (1999) Neural Networks, *2nd edition (Prentice Hall)*

Previous slide.

Soft clustering: Gaussian Mixture Model

20 centers per digit after clustering

(compare Perry Moerland 1999, Mixture model approach)



Previous slide.
In k-means clustering, a data point belongs exactly to one prototype (hard clustering).

Gaussian Mixture Models GMMs can be interpreted as ‘soft clustering’: Different protypes can take over partial responsibility for explaining a data point.

Here is a standard application: GMMs have been applied to model the distribution of all the handwritten digits ‘4’. What is shown is the center of each of the 20 Gaussians interpreted as pixels. The same is repeated for ‘1’ and ‘7’.

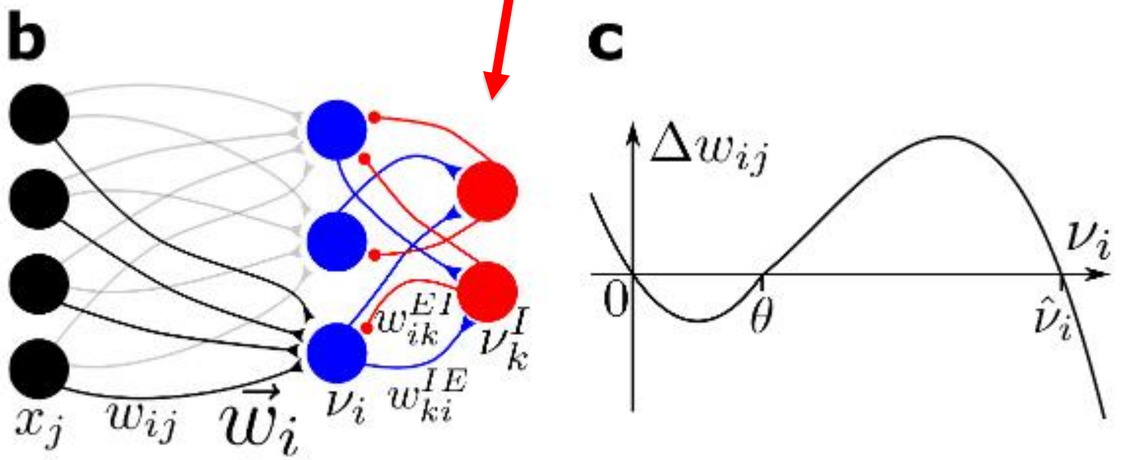
We do not cover the theory of GMMs here. Rather the point of the following two slides is to show that neuronal models with Hebbian rules can do something very similar to GMMs. If you are not familiar with GMMs you can forget this analogy and focus just on the next few slides.

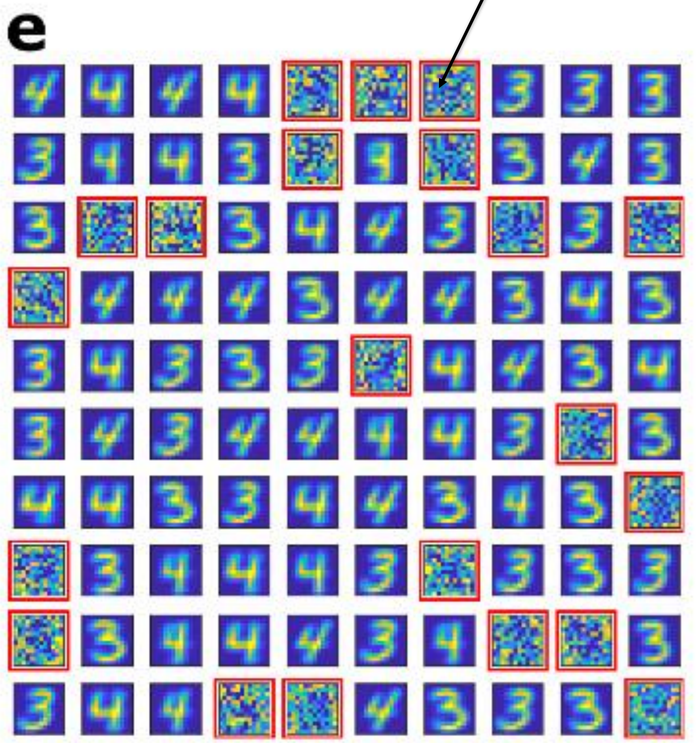
Weak competition caused by shared inhibitory neurons

inhibitory neuron

unspecific unit

Input:
MNIST 3 and 4.





synaptic plasticity (feedforward weights)

$$\Delta w_{ij} = a_2^{corr} \nu_j^{pre} \nu_i^{post} \left(\nu_i^{post} - \vartheta\right) - \left(\nu_i^{post}\right)^4$$

Gozel and Gerstner, 2021

Previous slide.

b) Real neurons in the brain are either excitatory (blue) or inhibitory (red). Therefore an excitatory neuron cannot simply inhibit another neuron, but inhibition arises indirectly via excitation of inhibitory neurons

c) The synaptic plasticity rule on the main slide is a bit simplified; the real plasticity rule is shown below. We can think of the term $(\nu_i^{post})^4$ as a generalized Oja-term.

e) MNIST images of 3 and 4 were applied at the input (black) while plasticity was on. After learning neurons either specialized on a specific writing style of 3 or 4, or remained unspecific. In the legend below, black neurons are called EC (entorhinal cortex) blue cells DGC (Dentat Gyrus Cells). Red cells are called GABAergic.

Fig 1a

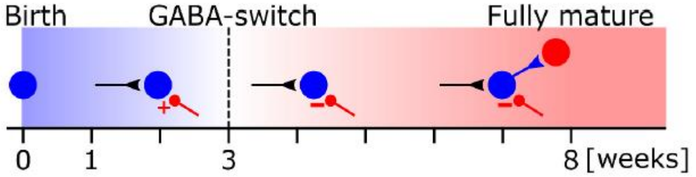
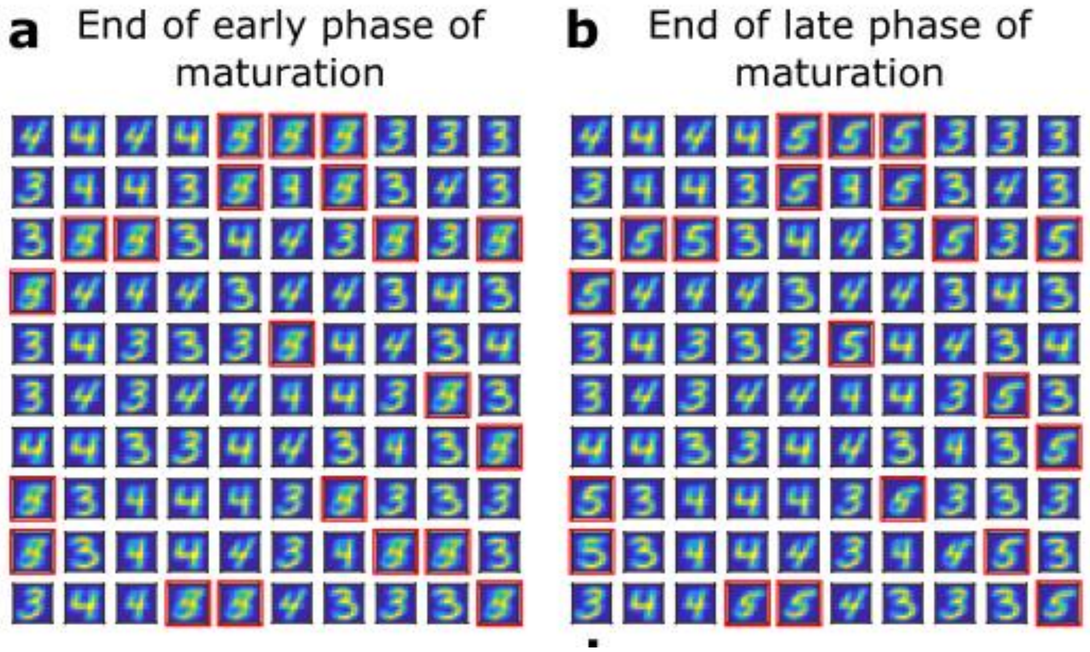
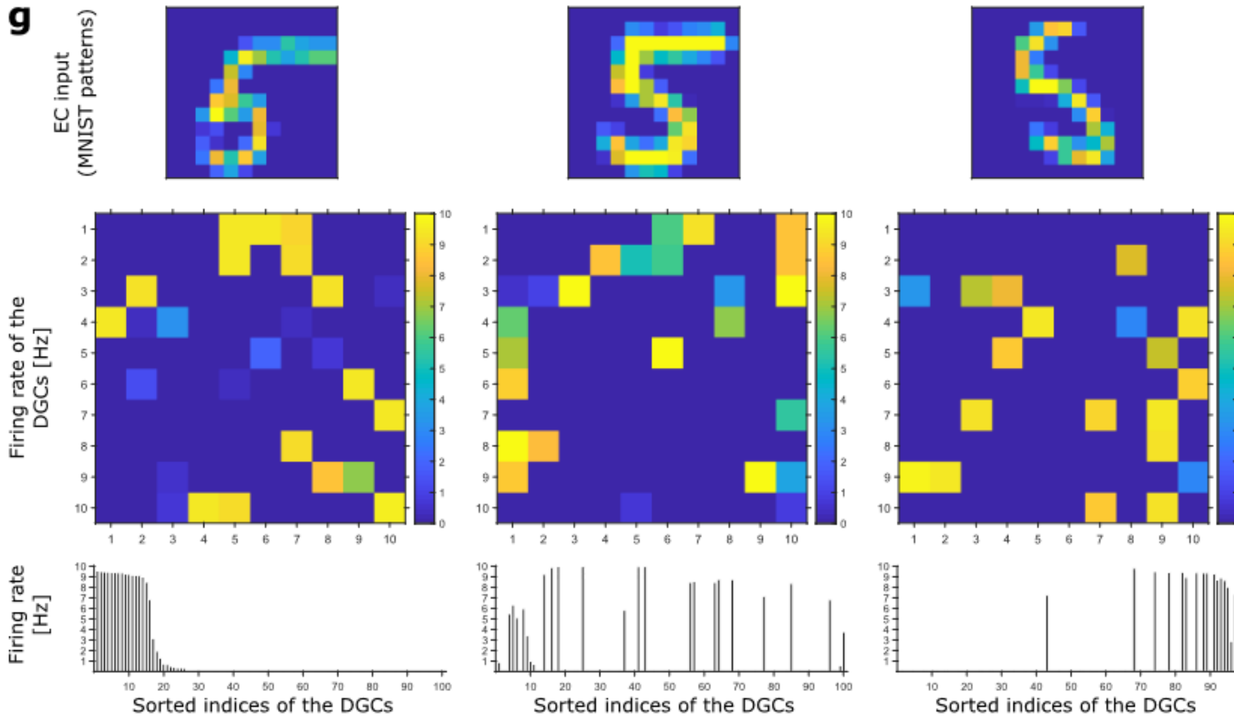


Fig 1c $\Delta w_{ij} = \eta \{\gamma x_j \nu_i [\nu_i - \theta]_+ - \alpha x_j \nu_i [\theta - \nu_i]_+ - \beta w_{ij} [\nu_i - \theta]_+ \nu_i^3\}$

Figure 1. Network model and pretraining. (**a**) Integration of an adult-born DGC (blue) as a function of time: GABAergic synaptic input (red) switches from excitatory (+) to inhibitory (−); strong connections to interneurons develop only later; glutamatergic synaptic input (black), interneuron (red). (**b**) Network structure. EC neurons (black, rate x_j) are fully connected with weights w_{ij} to DGCs (blue, rate ν_i). The feedforward weight vector $\vec{w}_i$ onto neuron i is depicted in black. DGCs and interneurons (red, rate ν_k^I) are mutually connected with probability p_{IE} and p_{EI} and weights w_{ki}^{IE} and w_{ik}^{EI}, respectively. Connections with a triangular (round) end are glutamatergic (GABAergic). (**c**) Given presynaptic activity $x_j > 0$, the weight update Δw_{ij} is shown as a function of the firing rate ν_i of the postsynaptic DGC with LTD for $\nu_i < \theta$ and LTP for $\theta < \nu_i < \hat{\nu}_i$. (**d**) Center of mass for three ensembles of patterns from the MNIST data set, visualized as 12 × 12 pixel patterns. The two-dimensional arrangements and colors are for visualization only. (**e**) One hundred receptive fields, each defined as the set of feedforward weights, are represented in a two-dimensional organization. After pretraining with patterns from MNIST digits 3 and 4, 79 DGCs have receptive fields corresponding to threes and fours of different writing styles, while 21 remain unselective (highlighted by red frames).

Soft competition: Each pattern represented by several neurons





Add Input: MNIST 5,
Present digits 3 and 4 and 5 in random order.
Initialize new neurons with small random weights.

Problem of ‘dead units’ avoided:
- Lateral interactions switch from excitation to inhibition

Gozel and Gerstner, 2021

Previous slide.
The dentate gyrus (DG) is one of the very few brain regions where neurons are born during adulthood. It receives input from EC (entorhinal cortex) plus lateral input from GABAergic cells. Importantly, shortly after birth the GABAergic cells EXCITE the newborn neuron. As a consequence of this, they are not in competition with other neurons but cooperate with other neurons. Therefore the feedforward plasticity of the newborn neurons makes them respond to the ‘average’ stimulus that the other neurons like.
a) In the model, previously unspecific neurons have died and are replace by newborn neurons. After stimulation with digits 3 and 4 and 5, the newborn neurons respond at the end of this early phase to the ‘average stimulus’. B) Later the GABAergic cells become inhibitory. In this late phase, the newborn cells are in competition with the existing cells. Therefore they specialize on different writing styles of 5.

(a,b))

Figure 2. Newborn DGCs become selective for novel patterns during maturation. (**a**) Unselective neurons are replaced by newborn DGCs, which learn their feedforward weights while patterns from digits 3, 4, and 5 are presented. At the end of the early phase of maturation, the receptive fields of all newborn DGCs (red frames) show mixed selectivity. (**b**) At the end of the late phase of maturation, newborn DGCs are selective for patterns from the novel digit 5, with different writing styles.

(g)

At the end of the late phase of maturation, three different patterns of digit 5 applied to EC neurons (top) cause different firing rate patterns of the 100 DGCs arranged in a matrix of 10-by-10 cells (middle). DGCs with a receptive field (see **b**) similar to a presented EC activation pattern respond more strongly than the others. Bottom: Firing rates of the DGCs with indices sorted from highest to lowest firing rate in response to the first pattern. All three patterns shown come from the testing set, and are correctly classified using our readout network.

Summary: Lateral interactions and competitive learning

Interaction type:

- linear inhibitory interaction with Hebbian rule:
  - → PCA with several components
  - → ICA/sparse coding with several components
- winner-take all: strong (fixed) inhibitory interaction
  - → k-means clustering
  - → a single neuron wins the competition
  - → Hebbian learning for all neurons, but only winner changes
- soft winner take all
  - → a combination of several neurons gets active (‘wins’)
  - → similar to sparse coding with several components
  - → interpolation between ‘prototypes’
  - → similar to Gaussian mixture models

Previous slide. Summary

Literature:
Dentate gyrus: https://en.wikipedia.org/wiki/Dentate_gyrus

Hyvarinen and Oja (2000) Independent Component Analysis: Algorithms and Applications, *Neural Networks*
Hyvarinen and Oja (1998), Independent Component Analysis by general nonlinear Hebbian rules, *Signal Processing.*
Gozel and Gerstner (2021) A functional model of the adult dentate gyrus. *eLife*. DOI: https://doi.org/10.7554/eLife.66463 1 of

Brito and Gerstner (2016), Nonlinear Hebbian learning as a unifying principle, *PLOS Comput. Biol.* DOI:10.1371/journal.pcbi.1005070

The end
of this part

Learning in Neural Networks: Lecture 3B

Receptive fields, Hebbian rules and ICA of image patches

Wulfram Gerstner

EPFL, Lausanne, Switzerland

ICA on image patches

Receptive fields in neuroscience

Development of receptive fields by Hebbian learning

Bell and Sejnowski, 1997, The “independent components” of natural scenes are edge filters, *Vision research*

Hyvarinen and Oja (2000) Independent Component Analysis: Algorithms and Applications, *Neural Networks*

Hyvarinen and Oja (1998), Independent Component Analysis by general nonlinear Hebbian rules, *Signal Processing.*

Gerstner and Brito (2016), Nonlinear Hebbian learning as a unifying principle, *PLOS Comput. Biol.* DOI:10.1371/journal.pcbi.1005070

Hubel and Wiesel (1962) , Receptive fields, binocular interaction and functional architecture in the cat's visual cortex *J. Physiol.* doi: 10.1113/jphysiol.1962.sp006837

Willshaw and von der Malsburg (1976) How patterned neural connections can be set up by self-organization, *Proc. Roy. Soc. Lond. B*. https://doi.org/10.1098/rspb.1976.0087

Quiz: ICA. Learning in neural networks

Suppose that we have sequence of data points $\vec{x}(t_1)$, $\vec{x}(t_2)$, $\vec{x}(t_3)$, … .

[ ] Spatial ICA requires data with a non-Gaussian distribution

[ ] Optimization of non-Gaussianity leads to different ICA rules, depending on the criterion of 'non-Gaussianity'

[ ] A Hebb-rule is an example of online rule

[ ] A non-Gaussianity function F leads to a Hebb-rule with nonlinearity F

[ ] A non-Gaussianity function F leads to a Hebb-rule with nonlinearity F'

[ ] FastICA implements an approximate Newton step for optimization

[ ] FastICA is a second-order gradient descent/ascent algorithm (i.e. includes curvature information of the loss/optimality criterion)

Review of lecture 2: ICA with Hebbian rules

Hebbian learning (2-factor rules) can be nonlinear

PCA can be derived from a maximization priciple

ICA and Blind Source Separation:
temporal and spatial ICA have different conditions

Optimization of Non-Gaussianity yields ICA via Hebbian rule
after transition from batch to online

ICA Algorithm FAST-ICA implements approximate Newton step

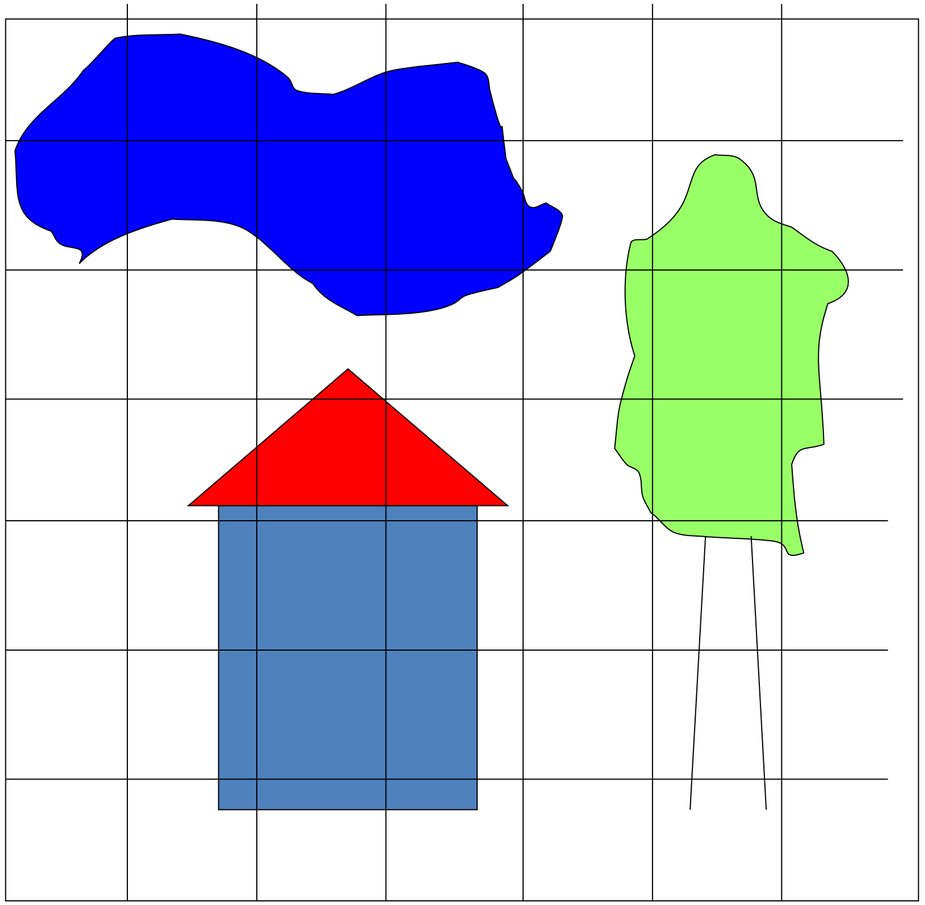
Previous slide.

In this section we ask two closely related questions:

1) What are the independent components of images?

2) What are receptive fields in visual cortex and how do they develop after birth?

What are the independent components of images?



Apply ICA algo
on image patches

Previous slide.

1) What are the independent components of images?

A common idea is that images are composed of objects and individual objects are composed of ‘image elements’. For example, a picture of a building is composed of many straight bars.

To find these image elements we apply ICA to thousands of image patches.

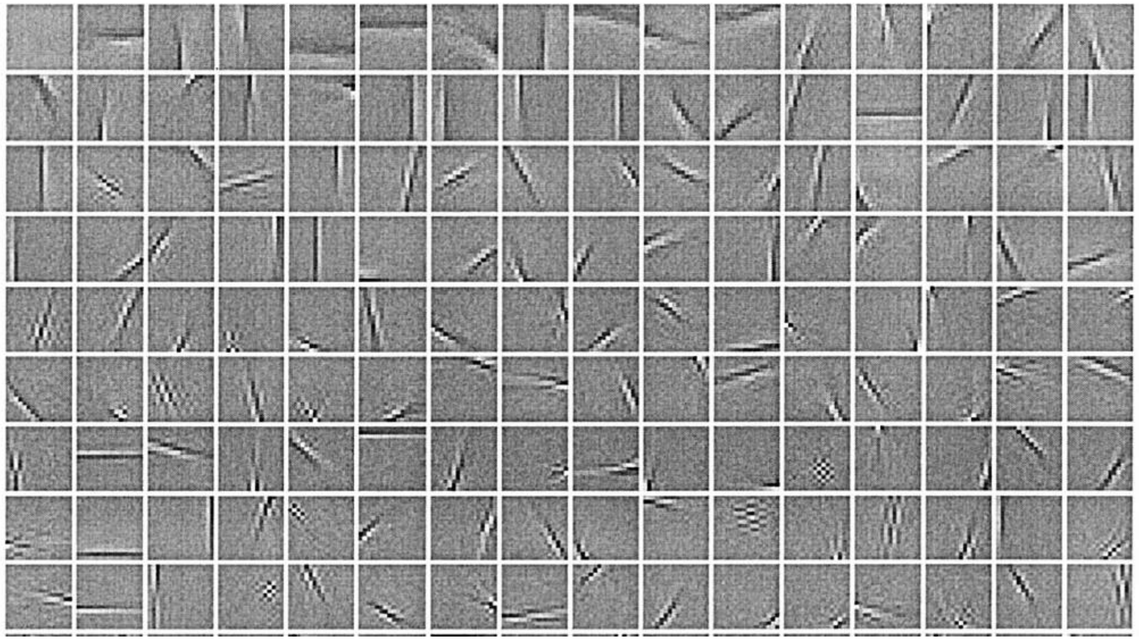
Figure 4: Basis functions in ICA of natural images. The input window size was 16 × 16 pixels. These basis functions can be considered as the independent features of images.

From: Hyvarinen and Oja, 2000;
See also: Bell and Sejnowski, 1997

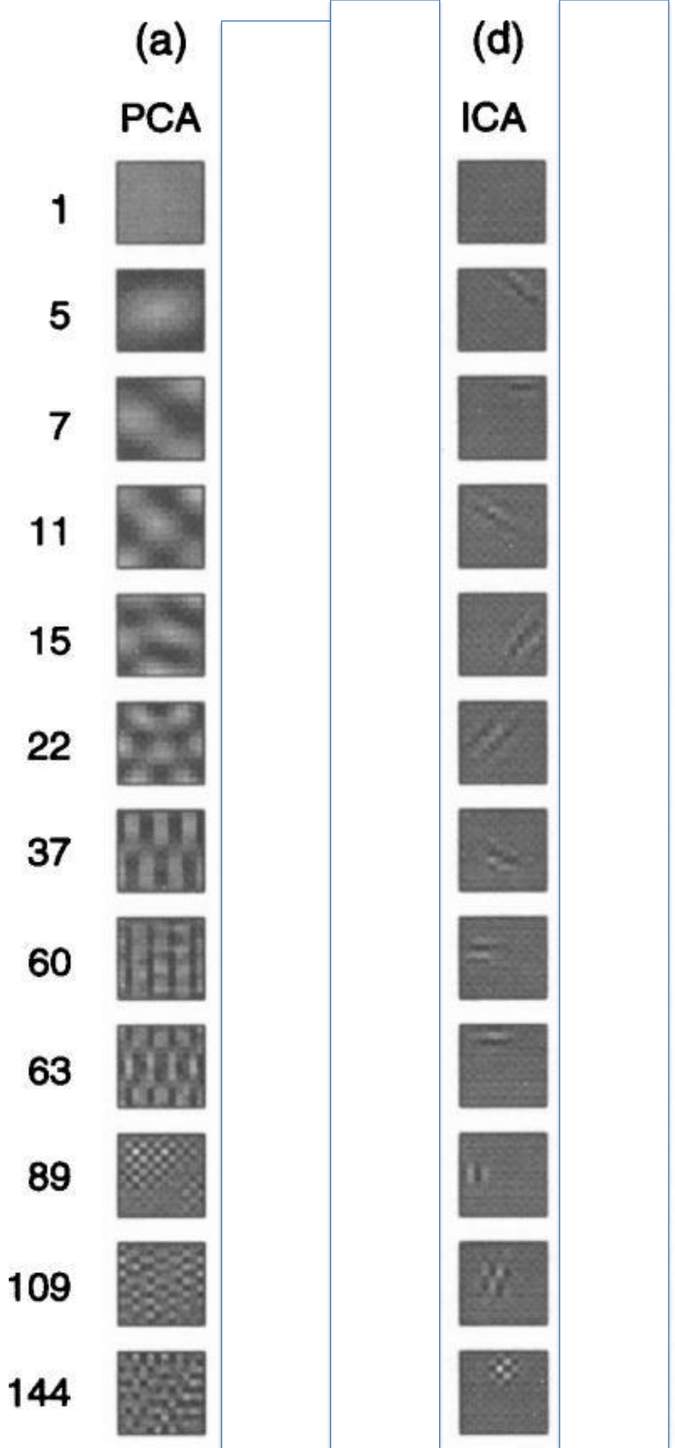
Previous slide.

ICA yields indeed many ‘bar-like’ elements.
On the left, a single independent component is shown. Below several.

These independent components can be interpreted as edge-detectors. Different ‘independent components’ correspond to edge detectors at different locations and with different orientations.



Independent components of images



Selected decorrelating filters and their basis functions extracted from natural scene data. Each type of decorrelating filter yielded 144 12 × 12 filters.

Bell and Sejnowksi,
Vision Research, 1997
The “independent components” of natural scenes are edge filters

Previous slide.
Compared to ICA, PCA typically yields more global components where each has structure that extends across the whole image patch. You can think of PCA as Fouriermodes of the two-dimensional patches. ICA, however, gives localized edge detectors.

Intermediate summary: ICA on image patches

1) What are the independent components of images?

→ ICA yields localized ‘edge detectors’.

2) How is this related to receptive fields?

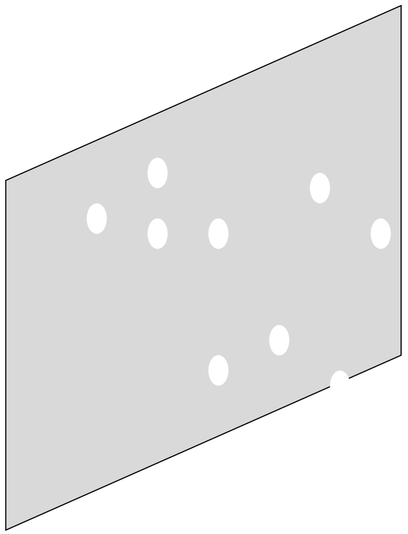
3) And how do receptive fields develop?

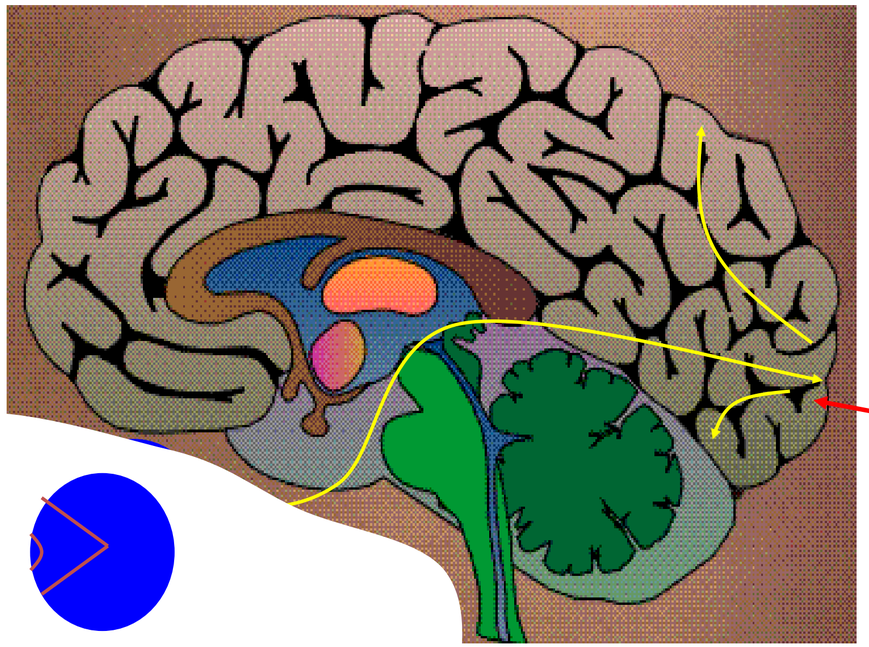
Previous slide.
First finding: ICA applied on image patches yields localized ‘edge detectors’

The second question now is:

How is this related to receptive fields? And how do receptive fields develop?

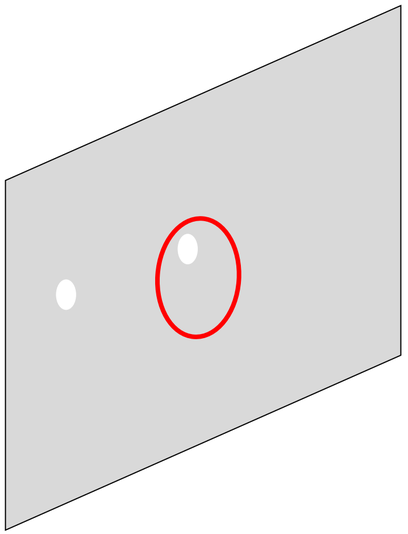
Receptive fields

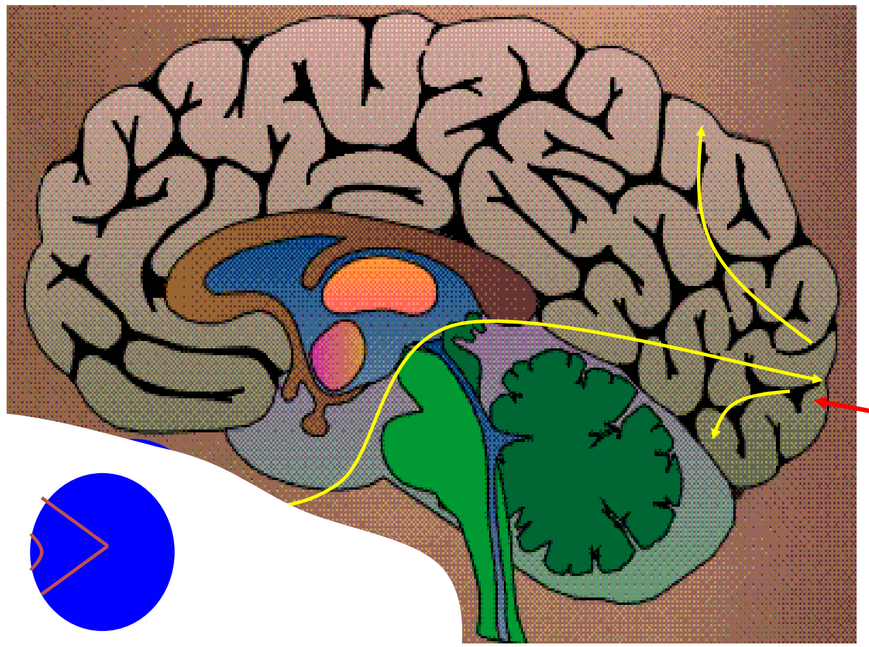




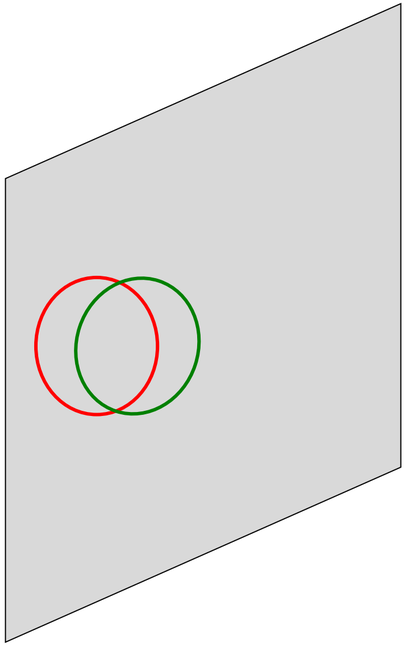
tok-tok-tok
tok-tok-tok

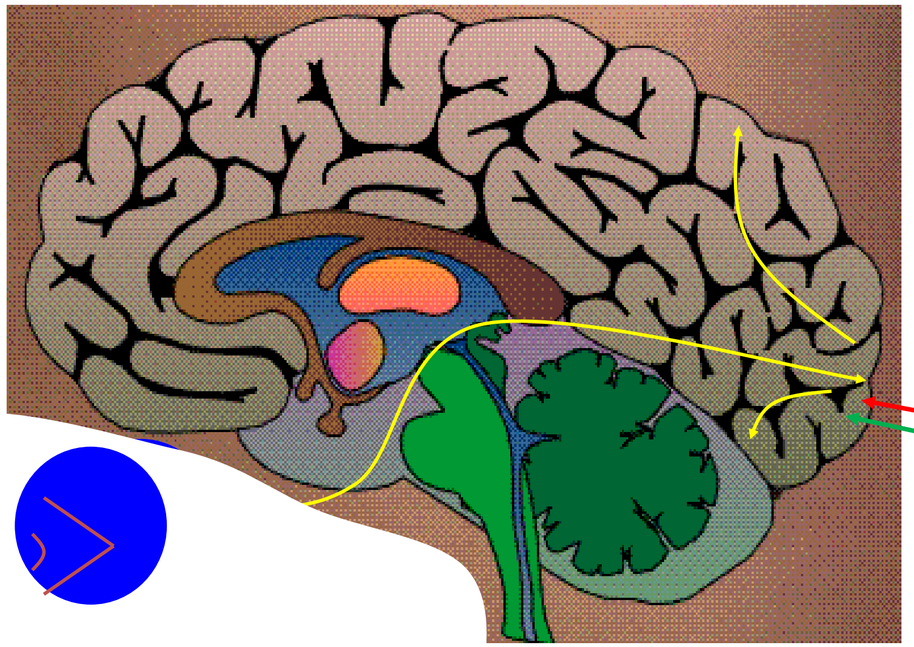
Receptive fields





Receptive fields and Retinotopic Map





visual cortex

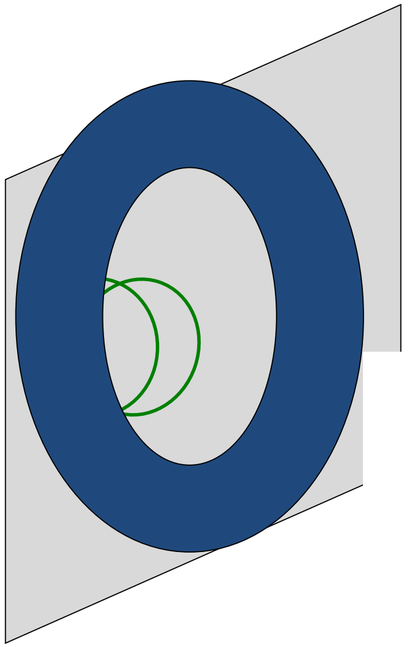
Neighboring cells
in visual cortex
have similar preferred
center of receptive field

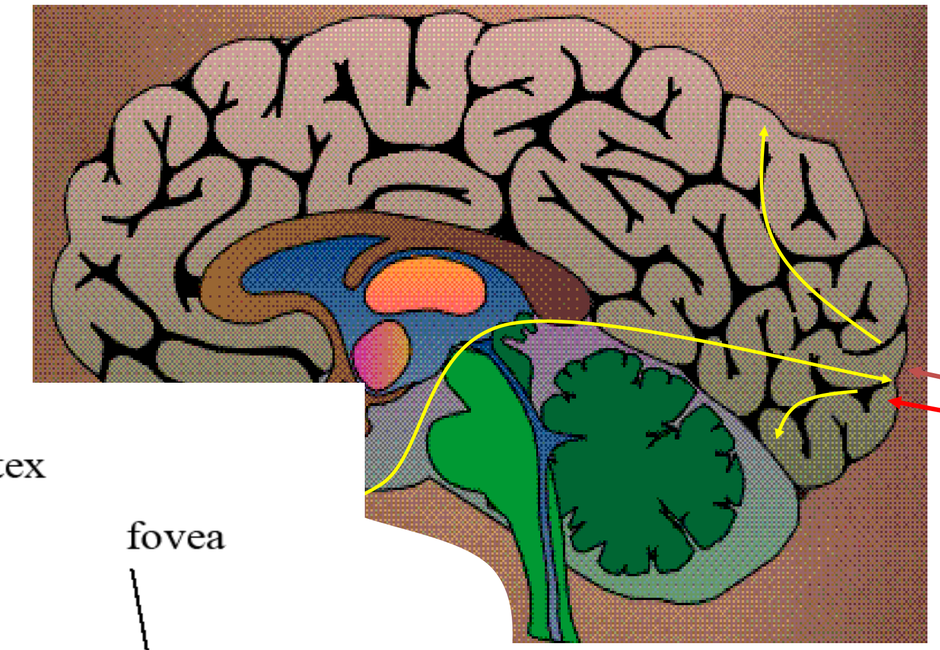
Previous slides.

Receptive fields of visual cells (in visual cortex area V1) are found as follows.

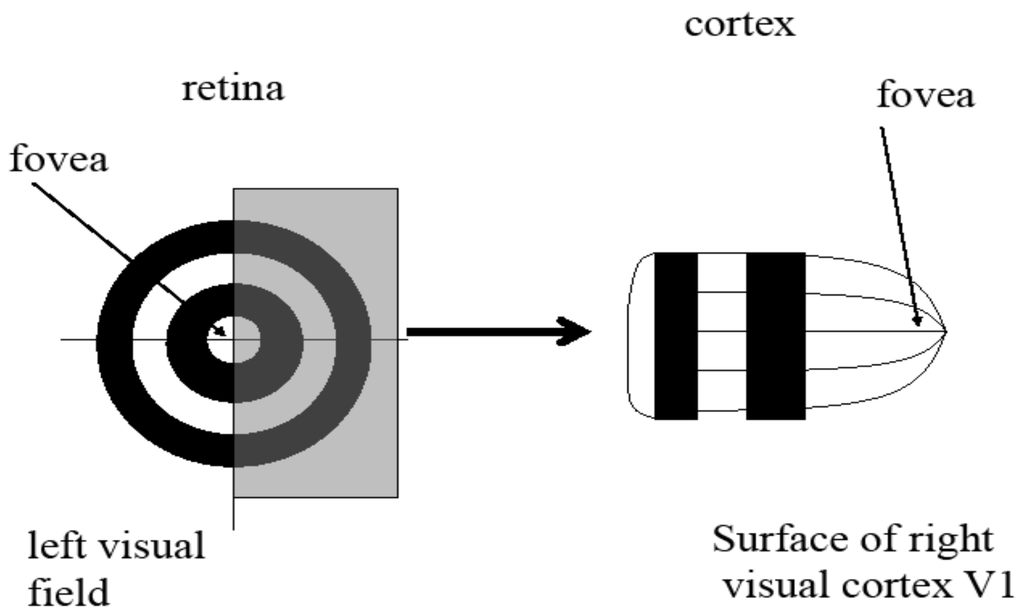
1. Insert an electrode close to one neuron. Put the electrical signal from the electrode on a loudspeaker.
2. Apply a single light dot at randomly selected locations on a gray visual screen.
3. The area of the visual screen on which a light dot causes electrical pulses (perceptible as tok-tok-tok from the speaker) is the visual receptive field of the cell.
4. You can test also with dark dots of light on the grey background; or by applying a prolonged light signal that is turned off after some time.

Receptive fields and Retinotopic Map





visual cortex



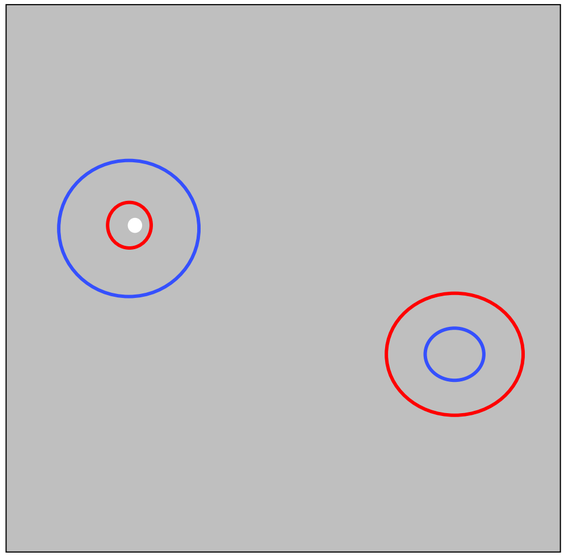
1. Neighboring cells in visual cortex have similar preferred center of receptive field
2. Globally a **'spatial map'** of outside world across V1

Hubel and Wiesel (1962) , Receptive fields, binocular interaction and functional architecture in the cat's visual cortex *J. Physiol.* doi: 10.1113/jphysiol.1962.sp006837

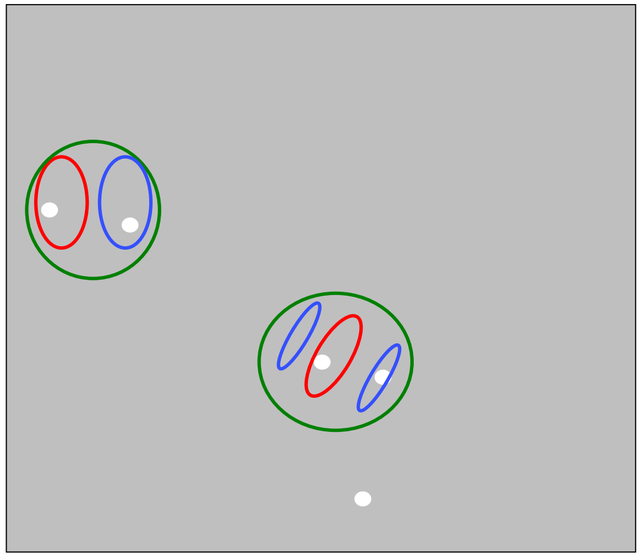
Previous slide.
Moreover, neighboring cells have similar receptive fields. Thus we have a map from screen location to location of neuron on the folded sheet of cortex. The map is distorted because the fovea takes much more space.

Receptive fields have a spatial structure

Receptive fields:
Retina, LGN



Receptive fields:
visual cortex V1



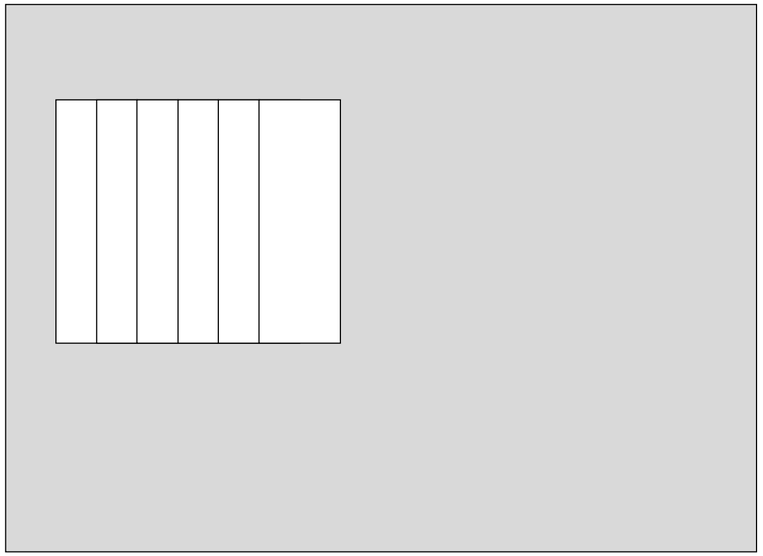
Orientation selective

Previous slide.
From the retina to cortex, signal transmission is recoded at an intermediate nucleus called LGN (lateral geniculate nucleus).
Cells in the LGN have circular receptive fields whereas cells in visual cortex V1 have elongated receptive fields. They are called orientation selective RFs (receptive fields).

Visual Cortex V1 Receptive fields show Orientation Tuning

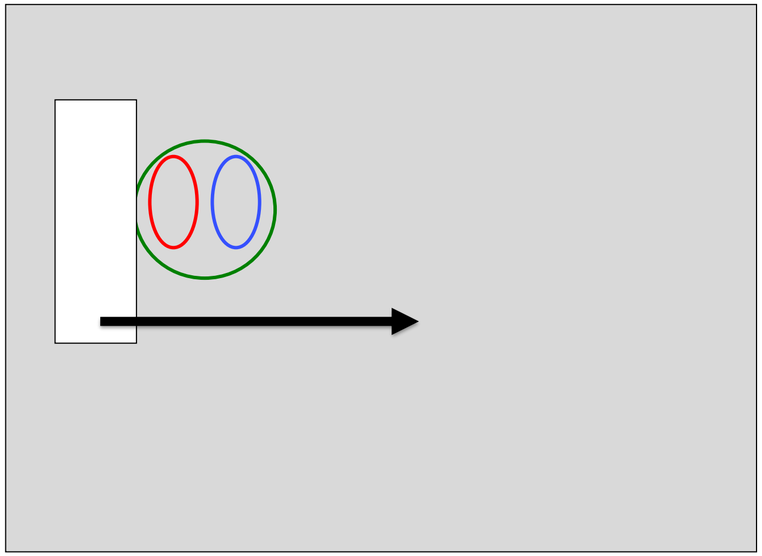
Receptive fields:

visual cortex V1

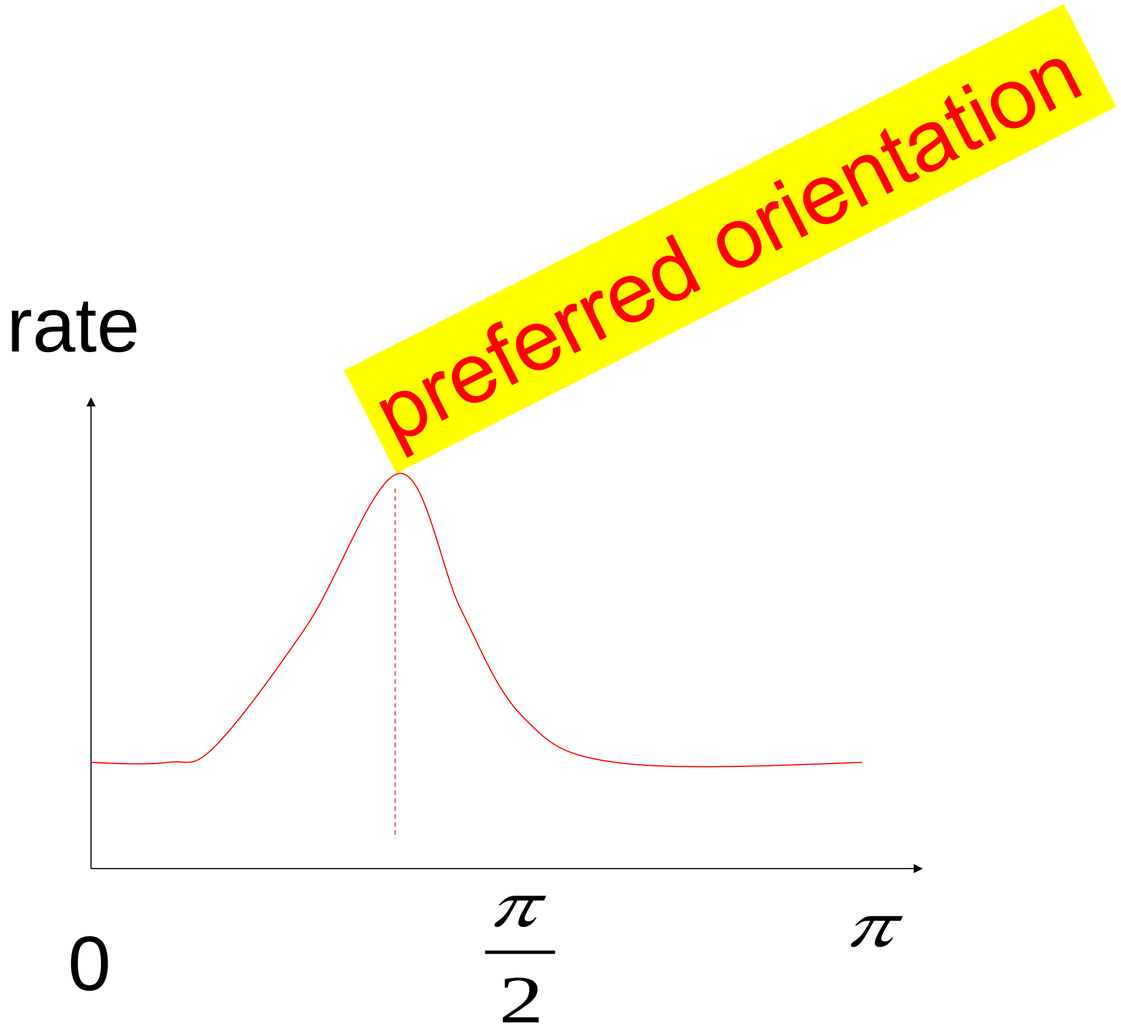


Orientation selective

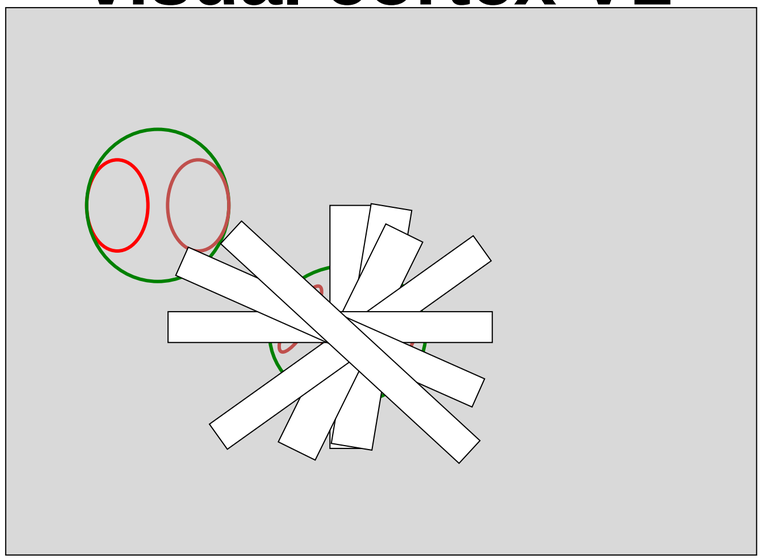
Previous slide.
Alternatively to a single light spot you can also stimulate with a slowly moving light bar.
You get maximal excitation of the neuron if the bar is aligned with the positive part (red) of the receptive field.



Receptive fields with Orientation Tuning



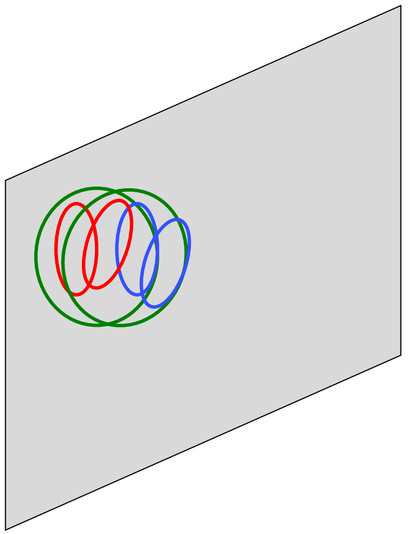
Receptive fields:
visual cortex V1

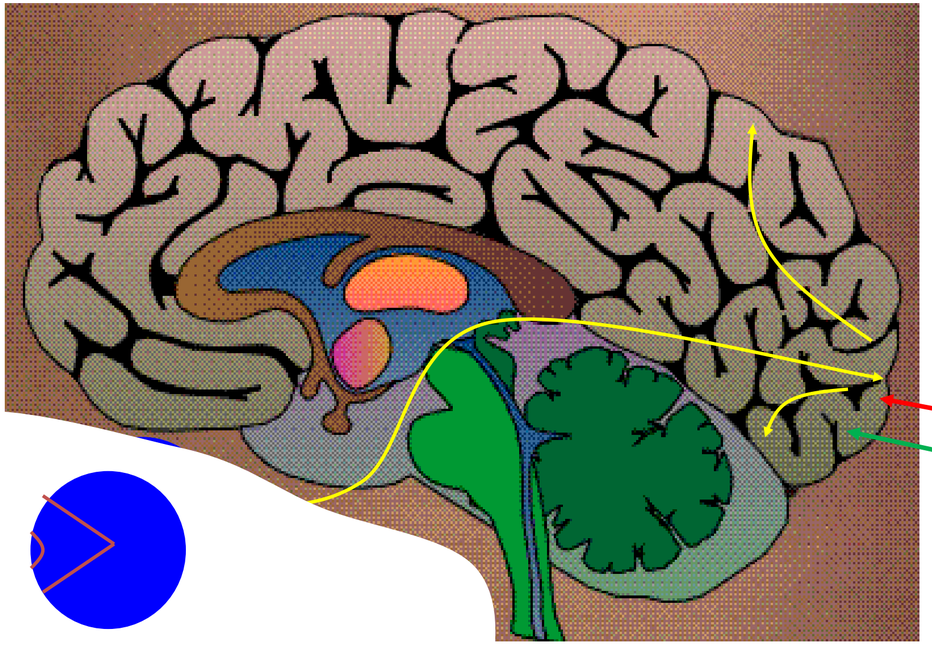


Orientation selective

Previous slide.
The term orientation selective arises because if you change the orientation of the light bar, there is a preferred orientation at which the neuron maximally responds.

Orientation Map





visual cortex

Neighboring cells in visual cortex
Have similar preferred orientation:

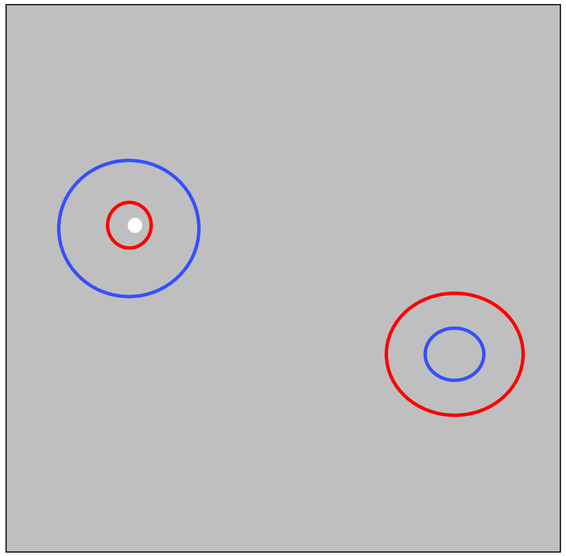
cortical orientation map

Hubel and Wiesel 1968; Bonhoeffer&Grinvald, 1991; Bressloff&Cowan, 2002; Kaschube et al. 2010

Previous slide.
Neighboring cells in cortex of cats and monkey also have similar preferred orientation.
The result is a cortical orientation map.

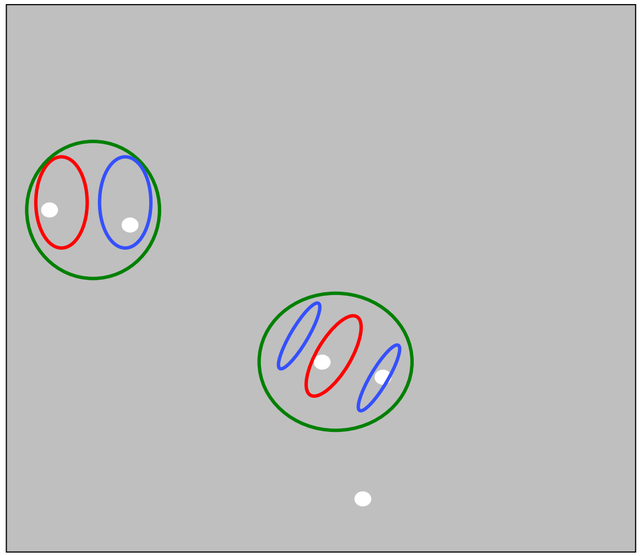
Receptive field Development

Receptive fields:
Retina, LGN



rotational
symmetry

Receptive fields:
visual cortex V1



orientation
selective

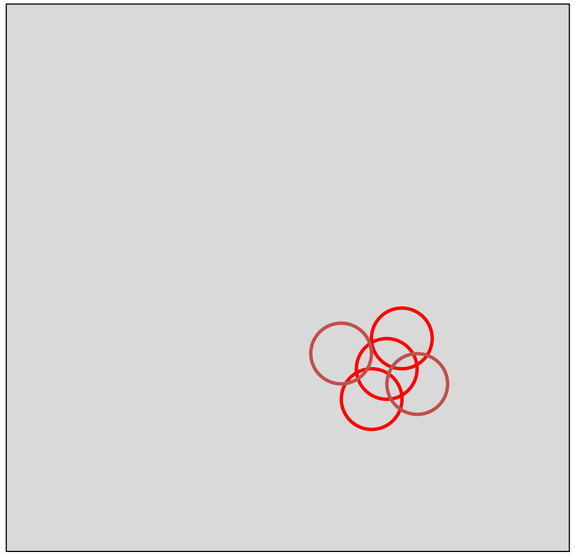
Previous slide.
From the retina to the LGN and then to cortex, information is preprocessed or ‘recoded’ as indicated by the different shapes of receptive fields. The elongated receptive fields with preferred orientation are useful as ‘edge detectors’ in V1 and in that sense potentially a ‘better code’.

The question then arises how this recoding arises.

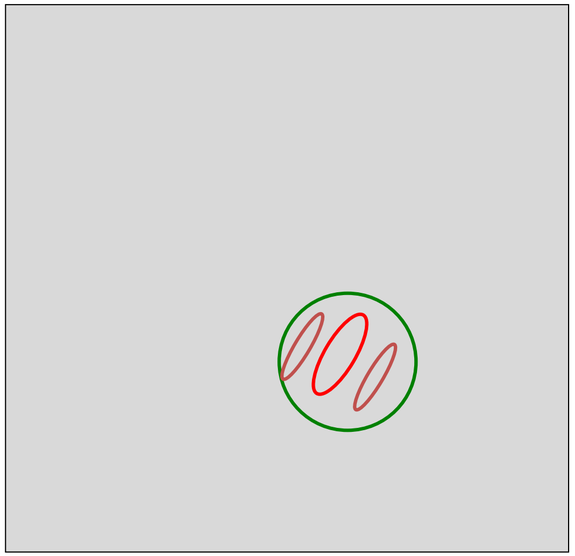
Receptive Field Development

What makes cells Orientation selective? – connectivity!

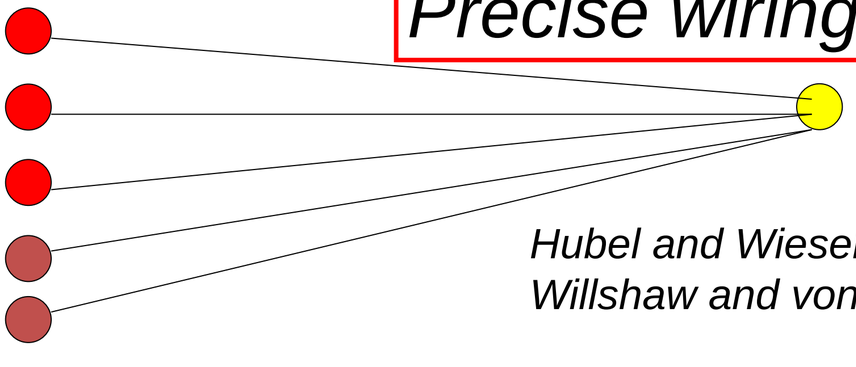
Receptive fields:
in LGN



Receptive fields:
visual cortex V1



Precise wiring necessary – how done?



Hubel and Wiesel (1962) , J. Physiol.
Willshaw and von der Malsburg (1976) Proc. Roy. Soc. Lond. B.

Previous slide.
The study of receptive fields in visual cortex of mammals by Hubel and Wiesel has been very influential. They proposed that the elongated receptive fields (edge detectors) arise by appropriate wiring of the connections to V1 arriving from LGN (the intermediate stop from the retina to cortex) .

But then the question arises, how such a precise wiring can develop.

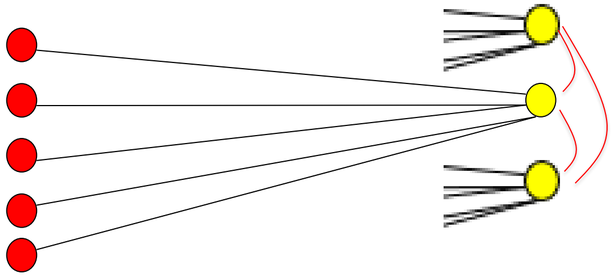
In particular, the wiring is not fixed but depends on the stimulation.
If young animals see only vertical stripes, they develop more edge detectors for vertical than for horizontal orientation.

From this and many other experiments can be concluded that the connections are not genetically encoded, but that the wiring depends on the statistics of stimulation. Now, this sounds like Hebbian learning could help to set up the wiring!

There have been many theoretical studies to illustrate how Hebbian learning could be used. The ideas can traced back at least to Willshaw and von der Malsburg (1976).

Apply nonlinear Hebbian learning rules/nonlinear neuron model

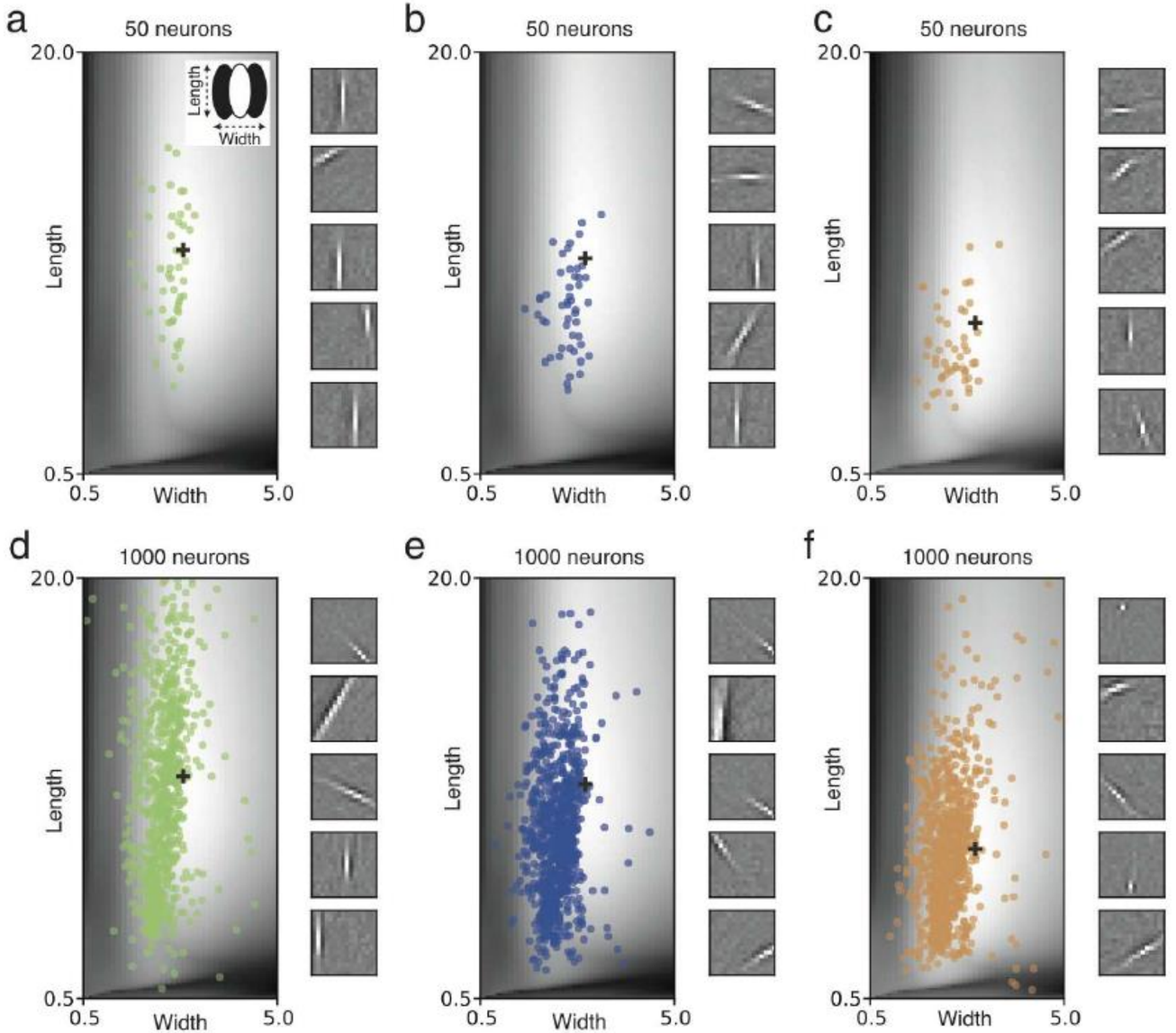
$$v_i^{post} = g(\sum_k w_{ik}\, v_k^{pre}) - \sum_n B_{in} v_n^{post}$$



-Inhibitory lateral interactions that learn with Hebbian rule

- 3 different types of nonlinearity g

All three nonlinearities lead to Gabor-like receptive fields!



Brito and Gerstner, 2016, PLOS comput. Biol.

Previous slide. *Gerstner and Brito (2016)*, Nonlinear Hebbian learning as a unifying principle, *PLOS Comput. Biol.* .

We focus on a recent study. It exploits that ICA arises from Hebbian learning in nonlinear neurons with gain function *g*. There are weak inhibitory interactions between visual cortex neurons (yellow). These interactions change according to the Hebbian learning rule that we have seen before.

$$\frac{d}{dt}B_{in} = +a^{lat}(\nu_i^{post} - \overline{\nu_i^{post}})\nu_n^{post}$$

The forward connections change according to the Hebb-rule combined with normalization. After presentation of pre-whitened image patches, learning results in elongated receptive fields (inset: schematic of Gabor filter) of each neuron. Different neurons have different RFs (5 samples shown). The diversity of RFs is shown as distribution of dots (one dot per neuron) indicating the width and length of the RF. Three different non-linearities (a,b,c) give very similar results.

Fig 4. Optimal receptive field shapes in model networks induce diversity. (**a-f**) Gray level indicates the optimization value for different lengths and widths (see inset in **a**) of oriented receptive fields for natural images, for the quadratic rectifier (left, see Fig 2a), linear rectifier (center) and L_0 sparse coding (right). Optima marked with a black cross. (**a-c**) Colored circles indicate the receptive fields of different shapes developed in a network of 50 neurons with lateral inhibitory connections. Insets on the right show example receptive fields developed during simulation. (**d-f**) Same for a network of 1000 neurons.

doi:10.1371/journal.pcbi.1005070.g004

Quiz

The receptive field of a visual neuron refers to

[ ] The localized region of space to which it is sensitive

[ ] The orientation of a light bar to which it is sensitive

[ ] The set of all stimulus features to which it is sensitive

The receptive field of a auditory neuron refers to

[ ] The set of all stimulus features to which it is sensitive

[ ] The range of frequencies to which it is sensitive

The receptive field of a somatosensory neuron refers to

[ ] The set of all stimulus features to which it is sensitive

[ ] The region of body surface to which it is sensitive

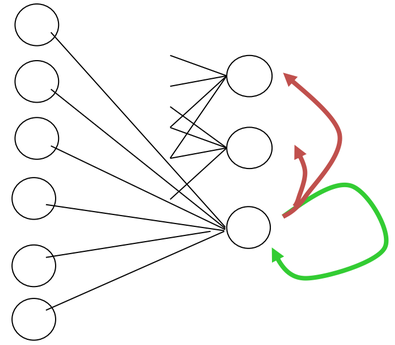
Previous slide.
The term ‘receptive field’ is also used outside vision.

Summary: ICA and Receptive Field Development

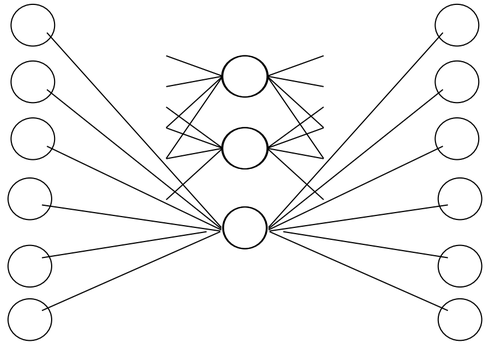
- Edge detectors are the independent component of image patches
- Edge detectors are typical for receptive fields in visual cortex V1
- ICA can be implemented by a nonlinear Hebbian learning rule
- Hebbian learning can explain the development of receptive fields in visual cortex V1 (and similarly in other primary sensory cortical areas)

Summary: the power of 2-factor rules

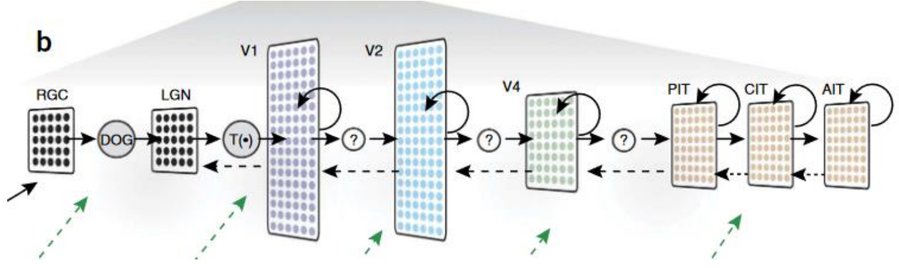
- 2-factor rules are Hebbian rules (‘pre’ and ‘post’).
- Hebbian rules have strong experimental support.
- 2-factor rules explain receptive field development
- 2-factor rules can implement **PCA**
- 2-factor rules can implement **ICA**
- 2-factor rules can implement **k-means clustering**.



- 2-factor rules can implement compressed **representation** for linear readout
- 2-factor rules can implement **autoencoders**
- BUT nearly always limited to 1 hidden layer



- Representation learning across multiple layers is nearly impossible with Hebb rule



The end