Lecture 3:
Modelling Insect Olfaction

Dr. Thomas Nowotny
University of Sussex

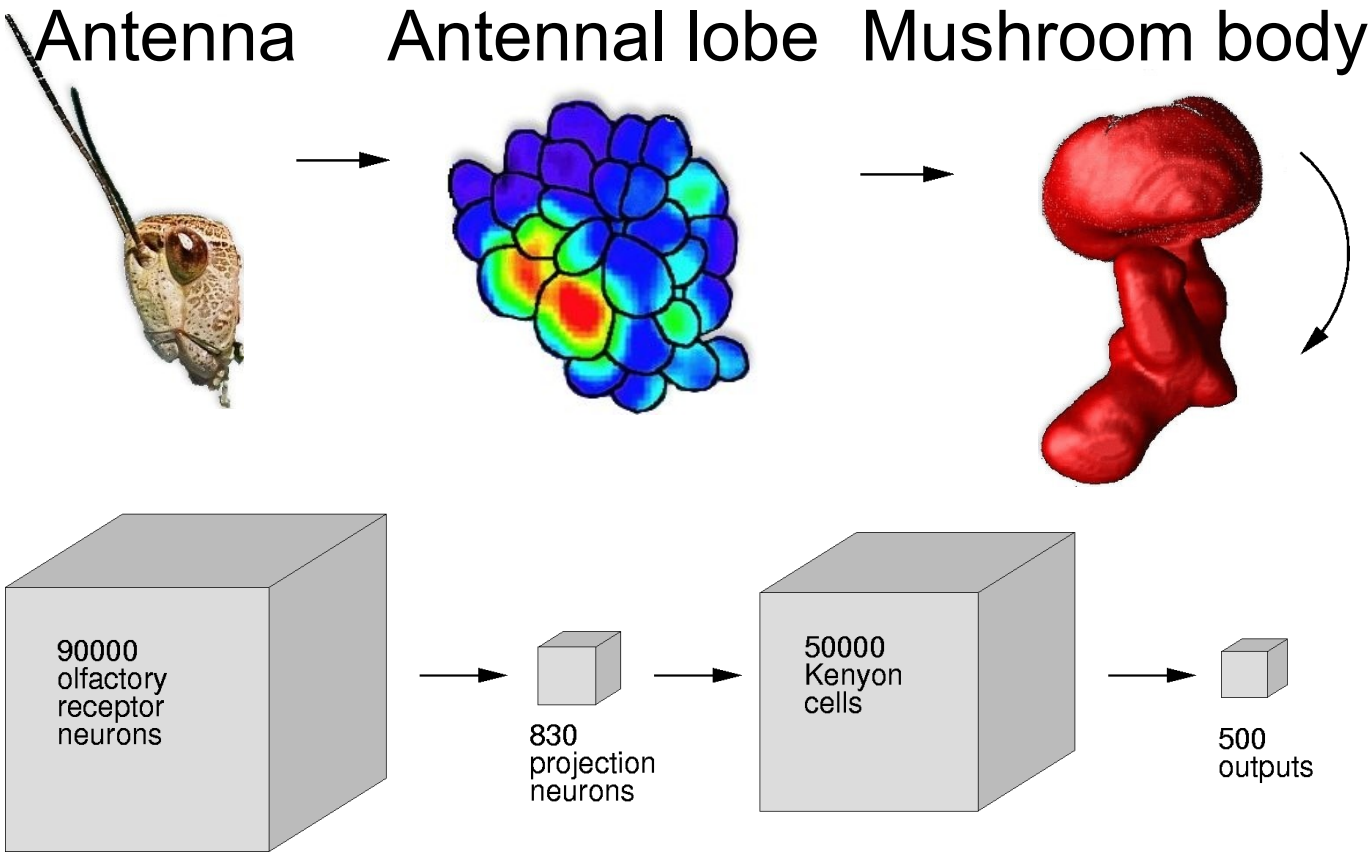
Models in insect olfaction

- Bazhenov et al. *Model of cellular and network mechanisms for odor-evoked temporal patterning in the locust antennal lobe* Neuron, 2001, 30, 569-581
- Bazhenov et al. *Model of transient oscillatory synchronization in the locust antennal lobe* Neuron, 2001, 30, 553-567
- Linster et al. *Computational diversity in a formal model of the insect olfactory macroglomerulus* Neural Comput, 1993, 5, 228-241
- Linster & Smith *Computational model of the reponse of honey bee antennal lobe circuitry to odor mixtures: Overshadowing, blocking and unblocking can arise from lateral inhibition* Behav Brain Res, 1997, 87, 1-14
- Galán et al. *Odor-Driven Attractor Dynamics in the Antennal Lobe Allow for Simple and Rapid Olfactory Pattern Classification* Neural Comput, 2004, 16, 999-1012

Models in insect olfaction

- Laurent et al. *Odor encoding as an active, dynamical process: Experiments, computation, and theory* Annu Rev Neurosci, 2001, 24, 263-297
- Huerta et al. *Learning classification in the olfactory system of insects* Neural Comput, 2004, 16, 1601-1640
- Nowotny et al. *Self-organization in the olfactory system: Rapid odor recognition in insects* Biol Cybern, 2005, 93, 436-446
- Nowotny et al. *Decoding temporal information through slow lateral excitation in the olfactory system of insects* J Comput Neurosci, 2003, 15, 271-281
- **And many others**

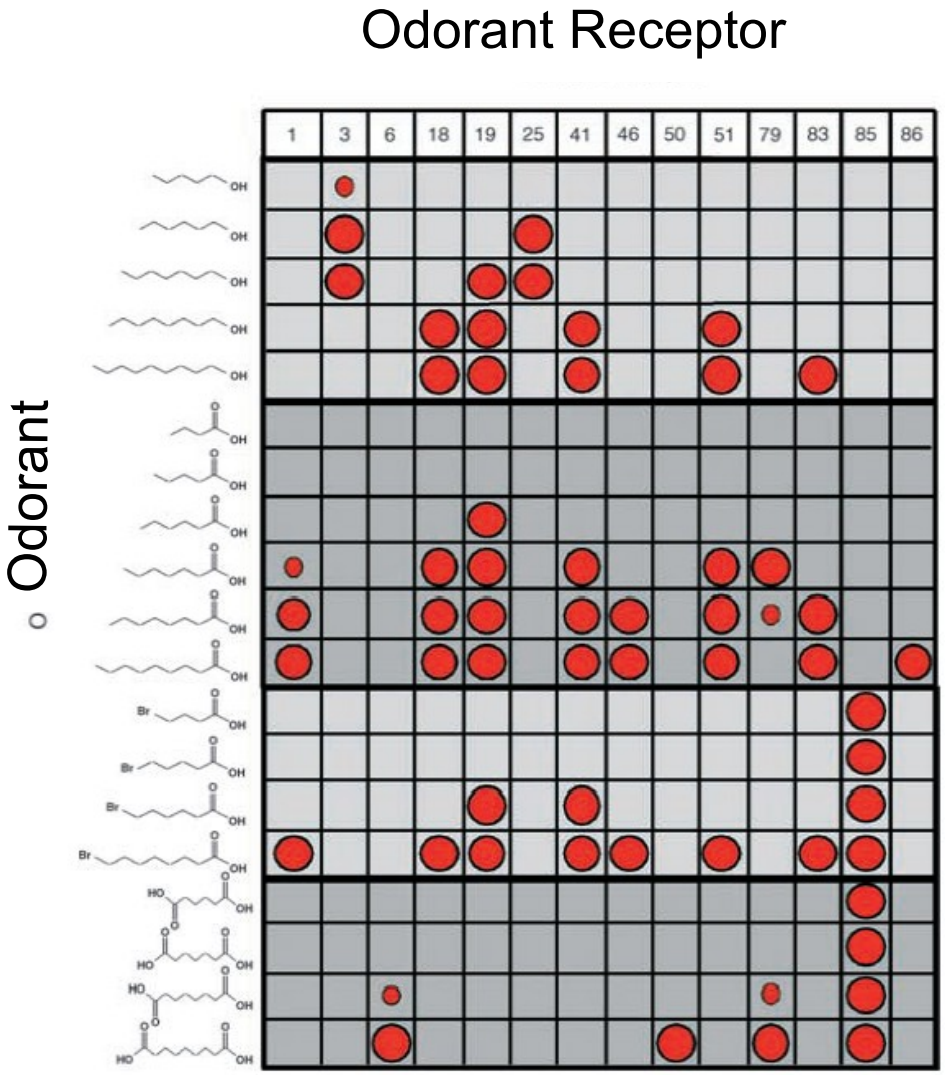
Reminder: Main olfactory pathway anatomy



Box volume ~ number of cells

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Olfactory Receptors



Odors evoke different, but overlapping patterns of receptor activity

From Linda Buck: Nobel lecture

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Early processing

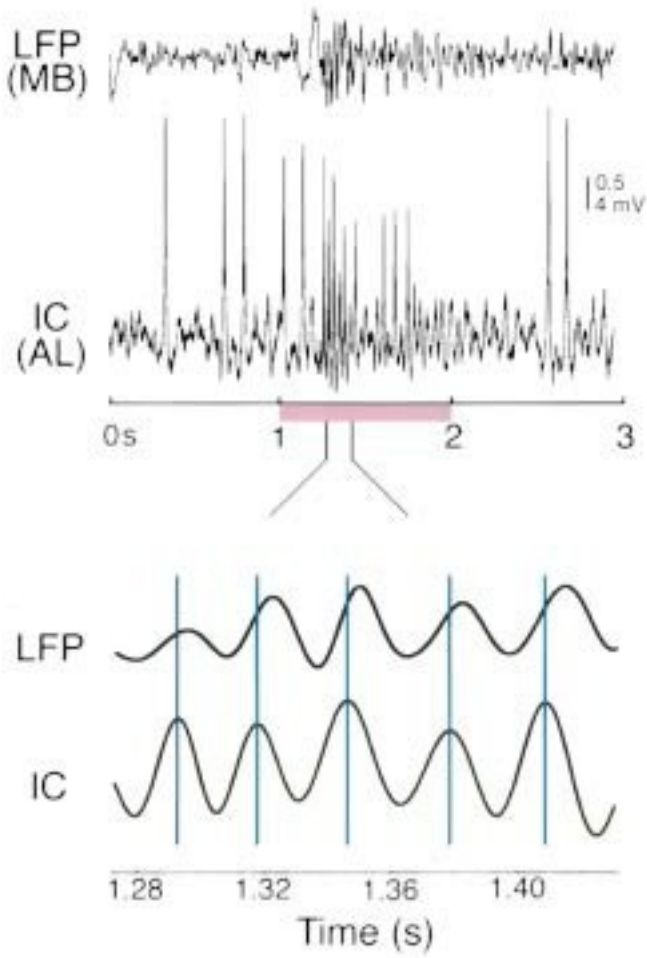
- Each olfactory receptor neuron expresses **one** receptor type
- All olfactory receptor neurons of the same type converge onto **the same** glomerulus
- Projection neurons receive inputs from **one** glomerulus

are encoded as overlapping patterns of projection neuron a

Local field potential

- There is a complex temporal dynamics in the AL
- One of the striking charact-eristics is the emergence of oscillations (measured in form of a local field potential (LFP):

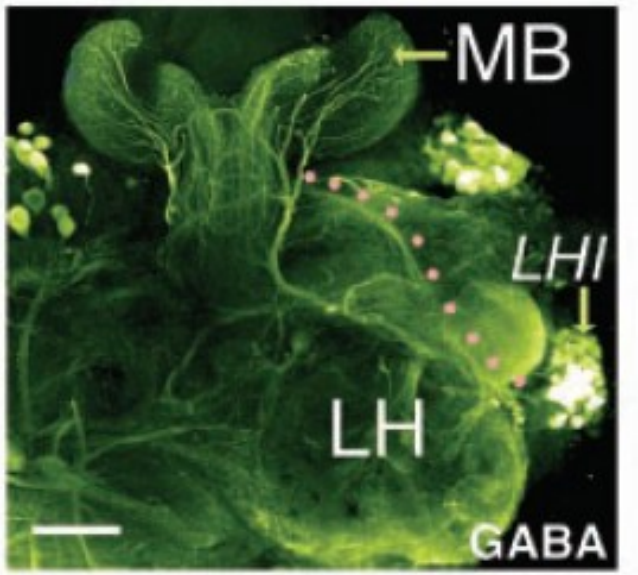
(we will talk about dynamics more in the next lecture)

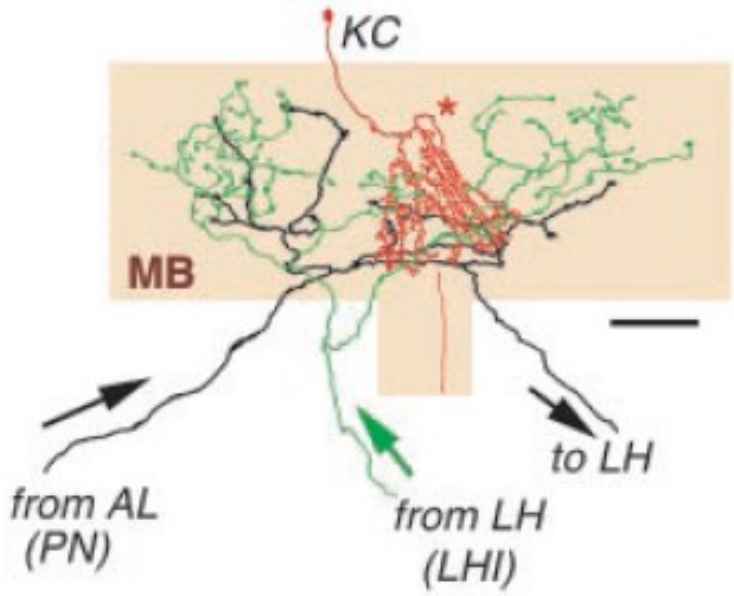


Stopfer et al.,
Nature 1997

Feedforward inhibition

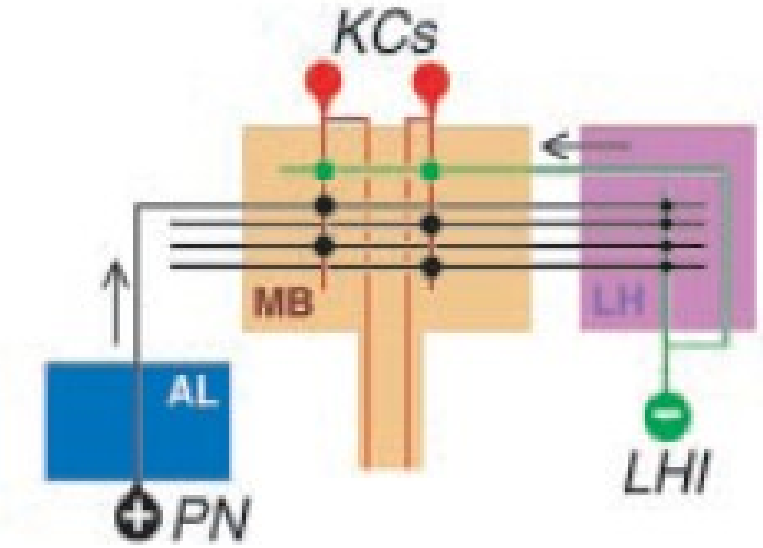
However, the complex activity in the AL is transmitted in a peculiar way:



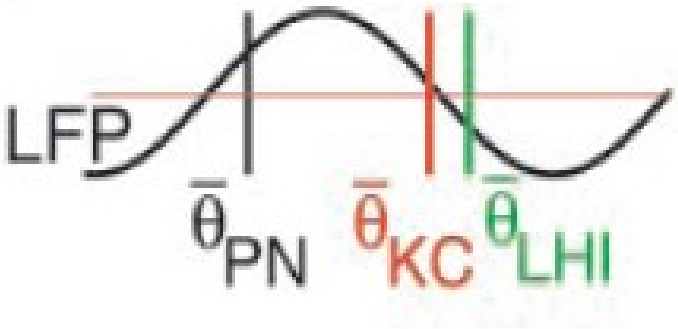


Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

AL dynamic patterns transmitted in "snapshots"



The Local Field Potential corresponds to a periodic 20 Hz inhibition onto KCs in the MB



Perez-Orive et al., Science (2002)

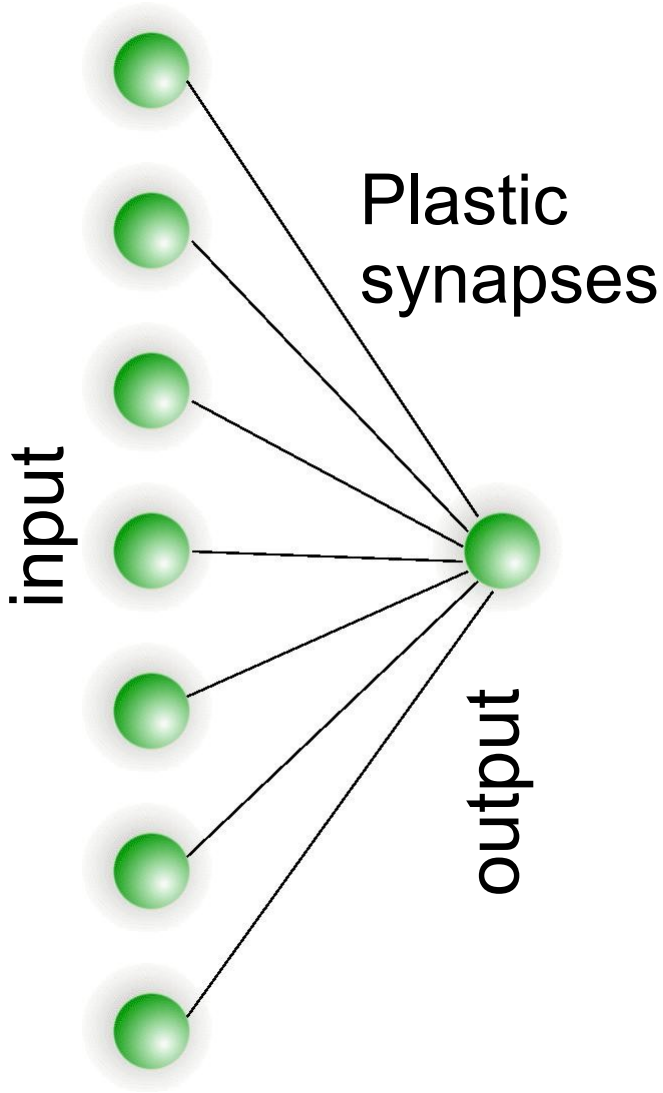
This "cuts the activity into snapshots".
These are likely processed separately (!)

Back to pattern classification

We have seen, that we can take the simple view for now that the task of the downstream system from the AL is the classification (recognition) of “snapshots” of activity patterns.

This is a classical task in machine learning/ artificial neural networks!

A classical pattern recognition solution



A simple perceptron rule:

Train A to respond to odor X (call it class 1)

… and hope that A does not respond to *any other odor*

(call it class -1)

What do you mean: “hope”?

McCulloch-Pitts neuron

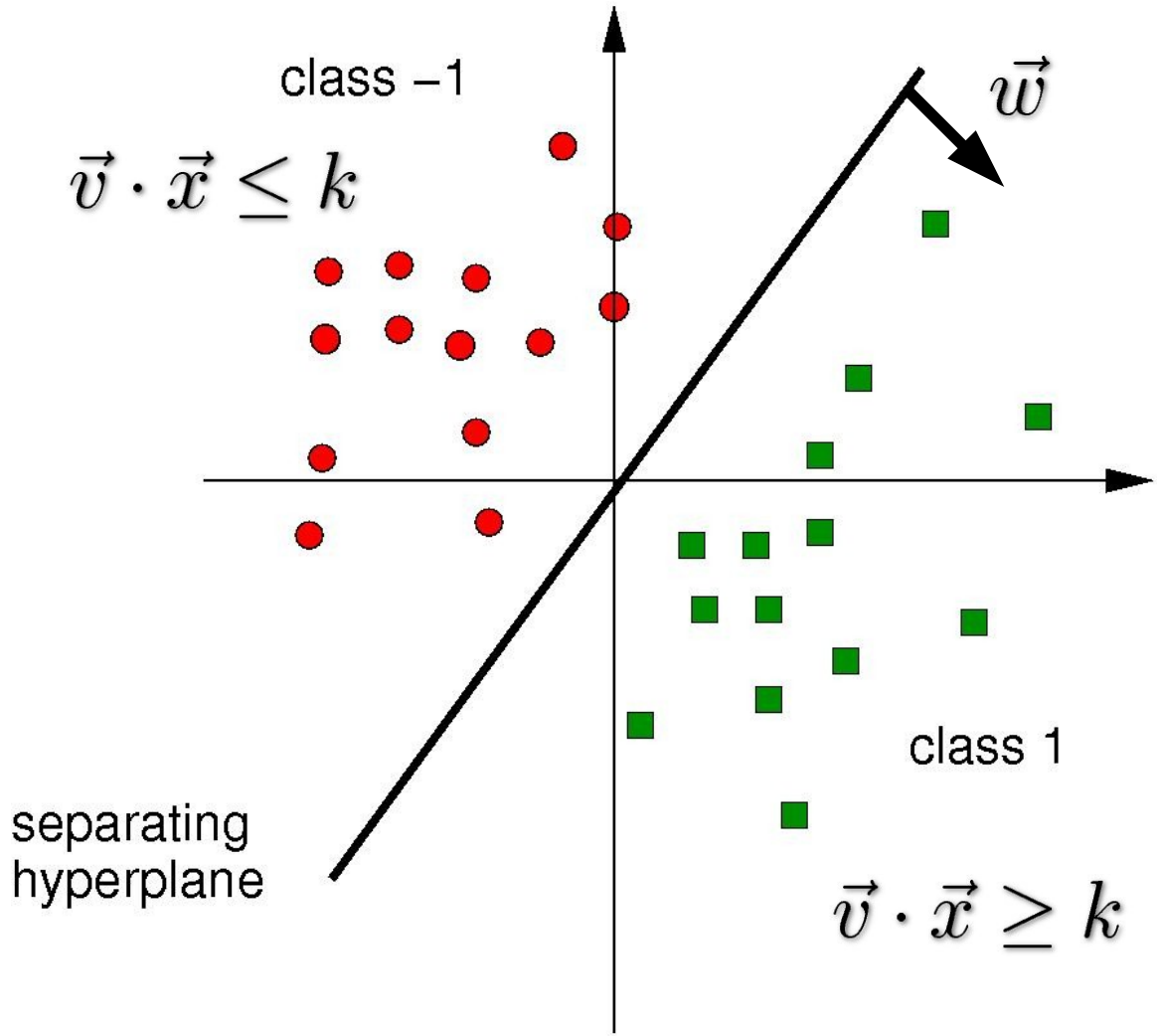
$$z(t) = \Theta\Big(\sum_j v_{kj} y_j(t-1) - \theta\Big)$$

“Hebbian” connections

$$v_{kj}(t) = \begin{cases} 1 & \text{with} \quad p_+ \quad \text{if } y_j = 1,\ z_k = 1 \\ 0 & \text{with} \quad p_- \quad \text{if } y_j = 1,\ z_k = 0 \\ v_{kj}(t-1) & \text{otherwise} \end{cases}$$

- Firing can be guaranteed for the right input.
- We have no control over the response to unknown inputs

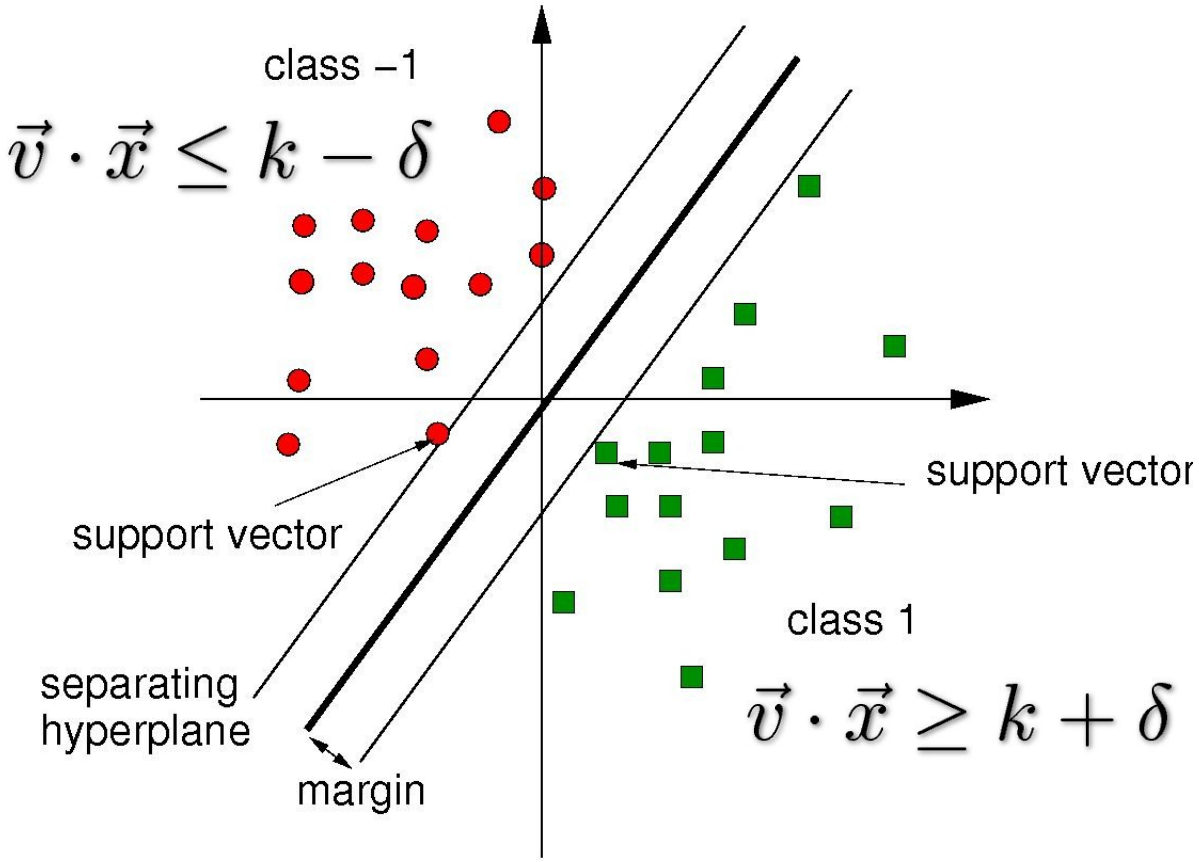
The perceptron is a linear classifier



The hyperplane is adjusted through the training and Hebbian learning

Support Vector Machines (SVM)

Cortes and Vapnik 1992,95: Support vector machine:

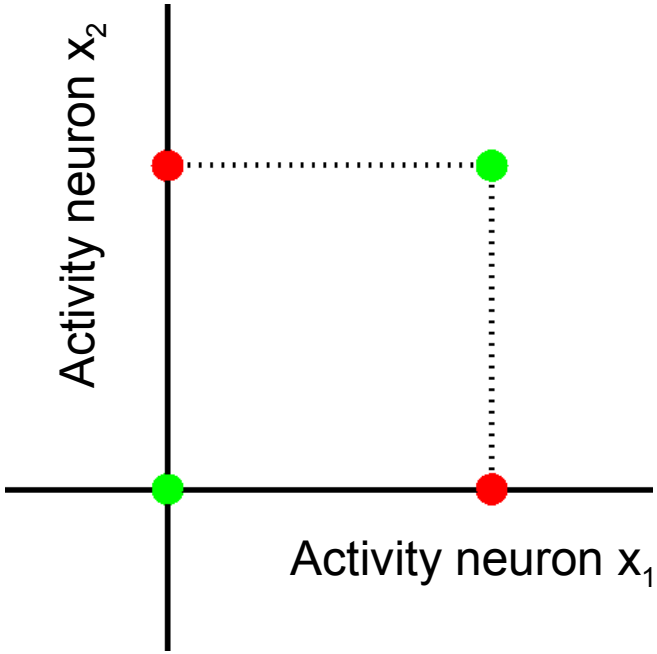


Here the hyperplane is adjusted to maximise the margin

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics

University of Sussex

Linear Classification can fail



There is no line that can separate green from red.

Dimension = number of neurons

Thomas Cover, 1965

"Classification is much more probable if the input is first cast into a high-dimensional space by a non-linear transformation."

Cover, T. (1965). Geometric and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE T Elect. Comput., 14, 326.

This can be done by using a non-linear "Kernel function" instead of the scalar product $\vec{w} \cdot \vec{x}$.
When used like this it is known as the "**kernel trick**".

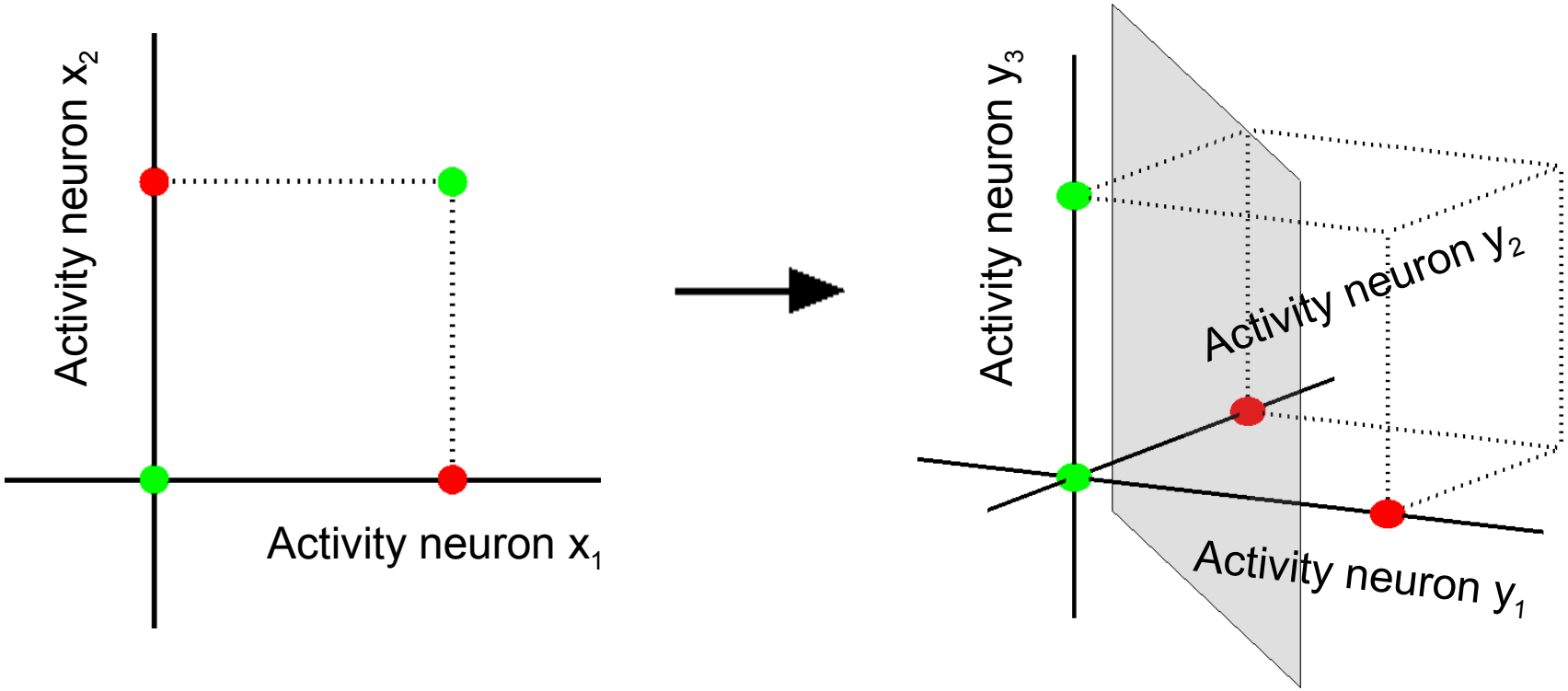
M. Aizerman, E. Braverman, and L. Rozonoer (1964). "Theoretical foundations of the potential function method in pattern recognition learning". Automation and Remote Control 25: 821–837

A related concept: MLP

If used with a large hidden layer, multi layer perceptrons (MLP) can also be seen as a related concept. The extra layer and nonlinear response of the neurons in it are the kernel/ nonlinear transformation.

See: F. Rosenblatt (1962) “Principles of Neurodynamics”. New York: Spartan books.

Nonlinear trafo / kernel / hidden layer "trick"



Dimension = number of neurons

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Typical kernels (transformations) used

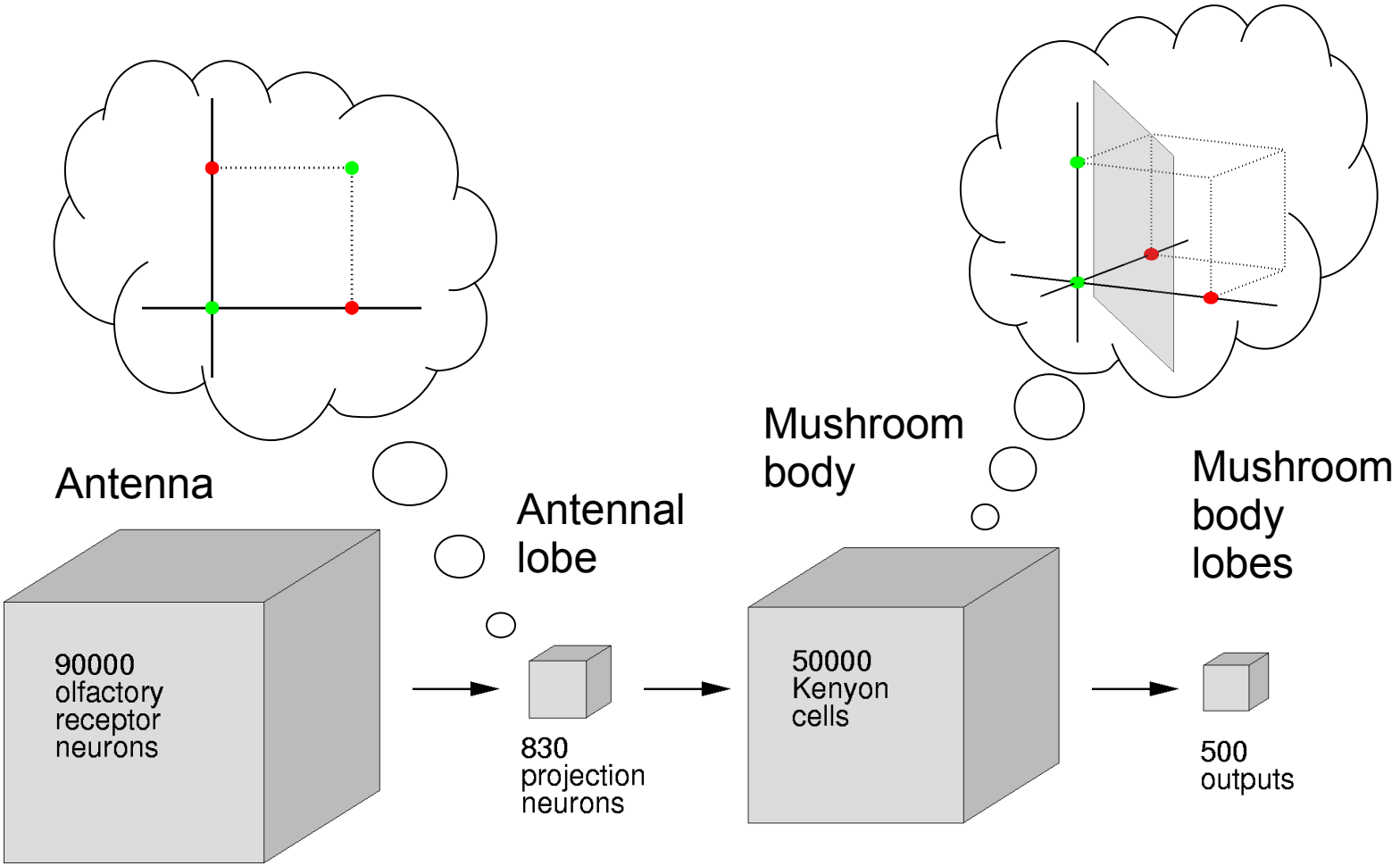
Polynomial (homogeneous): $K(\vec{w}, \vec{x}) = (\vec{w} \cdot \vec{x})^j$

Polynomial (inhomogeneous): $K(\vec{w}, \vec{x}) = (\vec{w} \cdot \vec{x} + 1)^j$

Radial Basis Function (general): $K(\vec{w}, \vec{x}) = K'(\|\vec{x} - \vec{w}\|)$

Gaussian RBF: $K(\vec{w}, \vec{x}) = \exp(-\gamma \|\vec{x} - \vec{w}\|^2)$

Hypothesis: The locust uses this idea



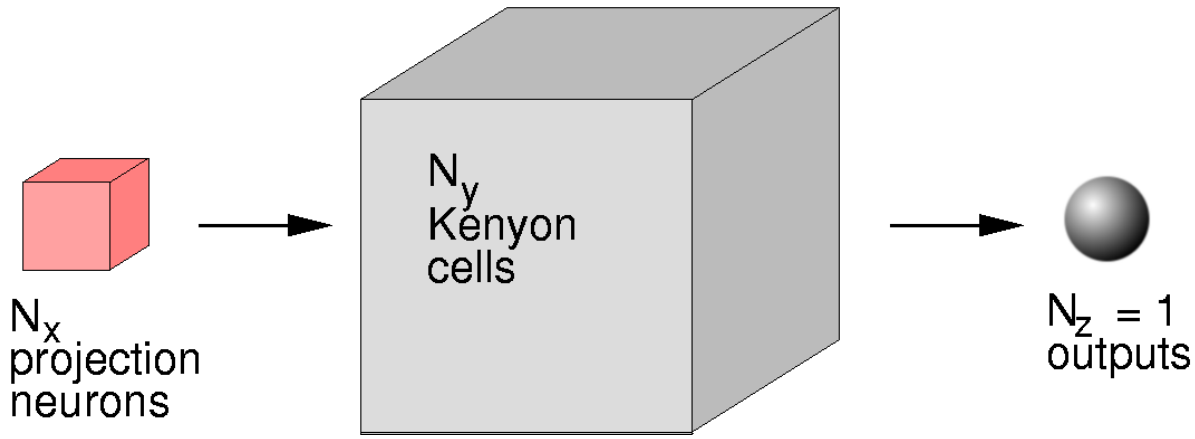
But we will use a *random kernel* (random connections)

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Classify one pattern from the rest

Random input patterns

$$x_i = \begin{cases} 1 & \text{with} \quad p_x \\ 0 & \text{otherwise} \end{cases}$$

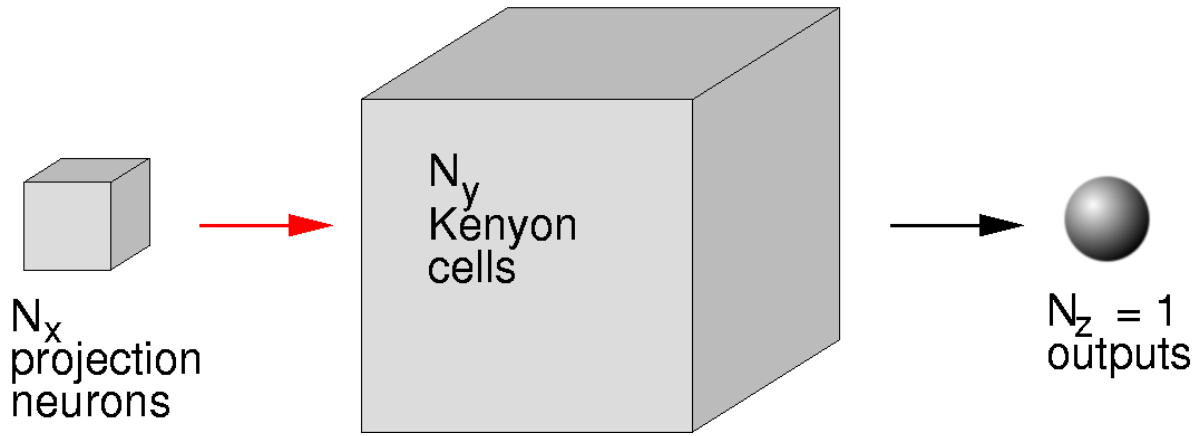


Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Classify one pattern from the rest

Random connections

$$w_{ji} = \begin{cases} 1 & \text{with} \quad p_{y \leftarrow x} \\ 0 & \text{otherwise} \end{cases}$$

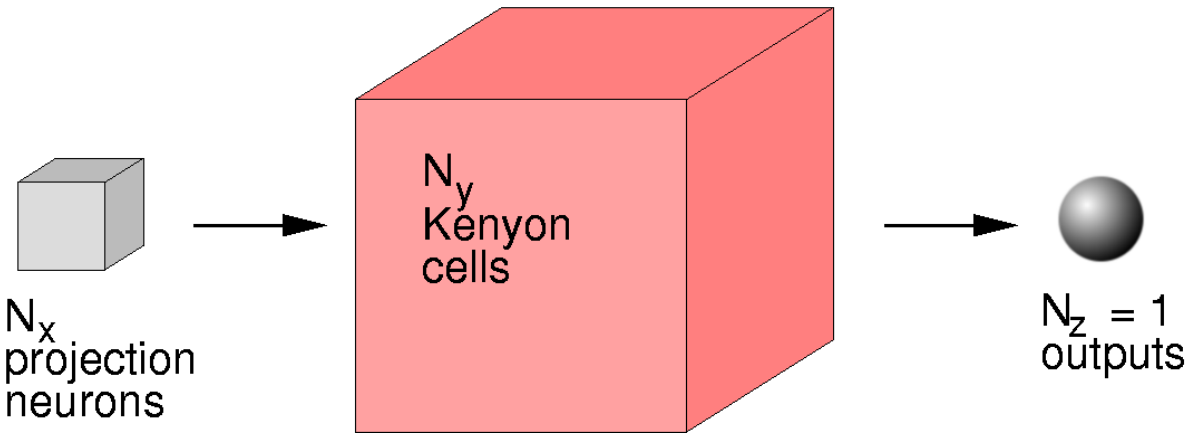


Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Classify one pattern from the rest

McCulloch-Pitts neurons

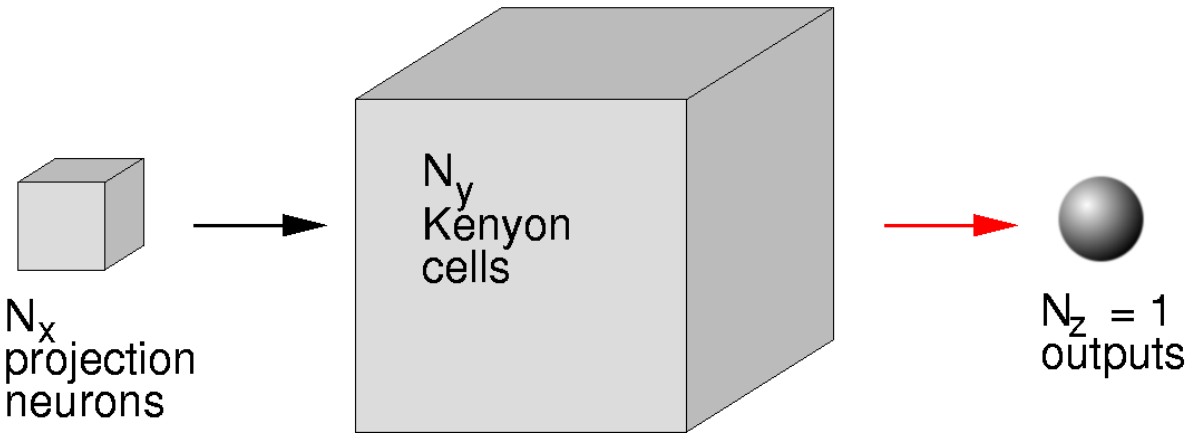
$$y_j(t) = \Theta\left(\sum_i w_{ji} x_i(t-1) - \theta\right)$$



Classify one pattern from the rest

"Hebbian" connections

$$v_{kj}(t) = \begin{cases} 1 & \text{with} \quad p_+ \quad \text{if } y_j = 1,\ z_k = 1 \\ 0 & \text{with} \quad p_- \quad \text{if } y_j = 1,\ z_k = 0 \\ v_{kj}(t-1) & \text{otherwise} \end{cases}$$

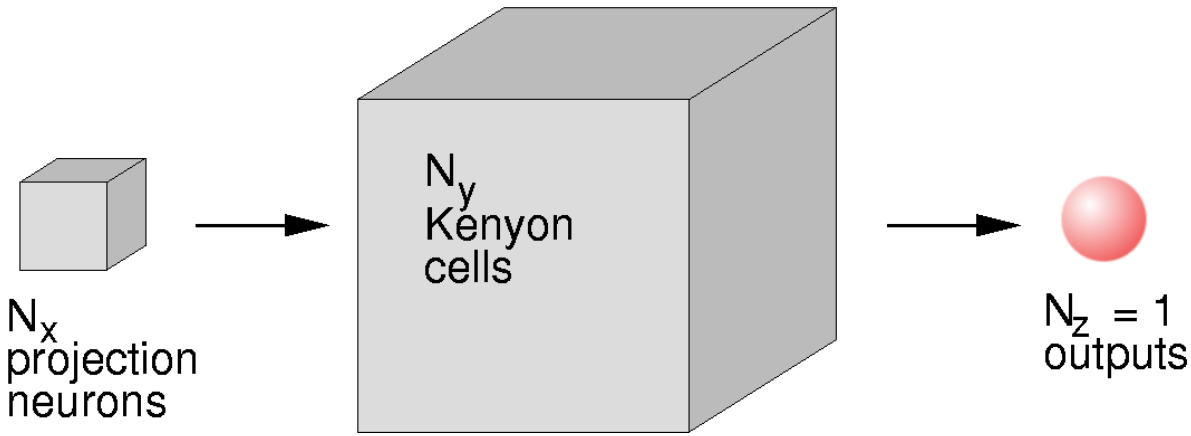


Classify one pattern from the rest

McCulloch-Pitts neuron

$$z(t) = \Theta\left(\sum_j v_{kj} y_j(t-1) - \theta\right)$$

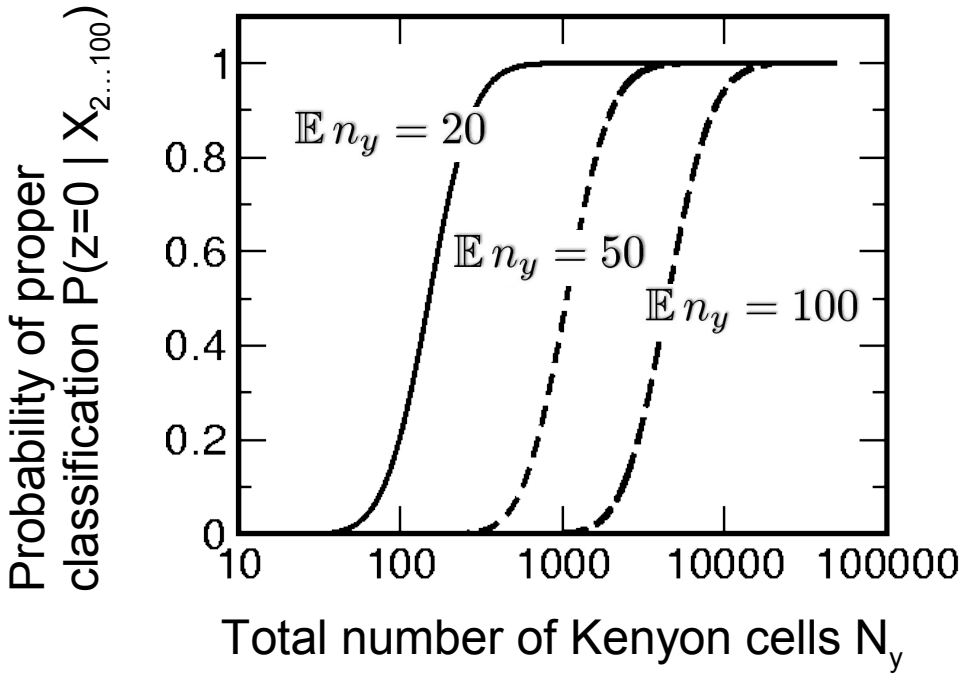
Induce a spike for 1 trained pattern
Don't do anything for 99 others

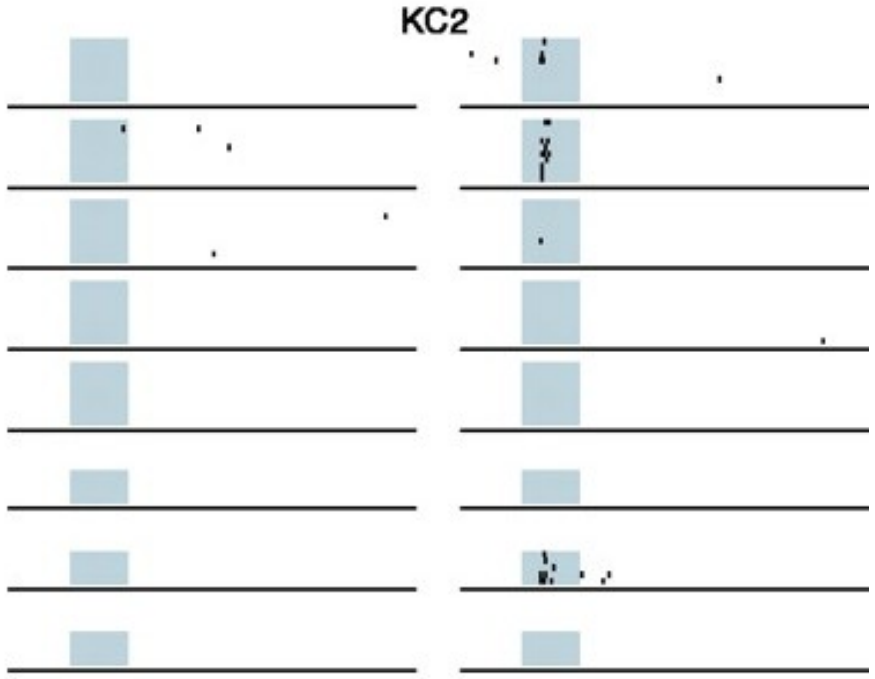


Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Example result: Classification needs sparse code

… and nature uses it!





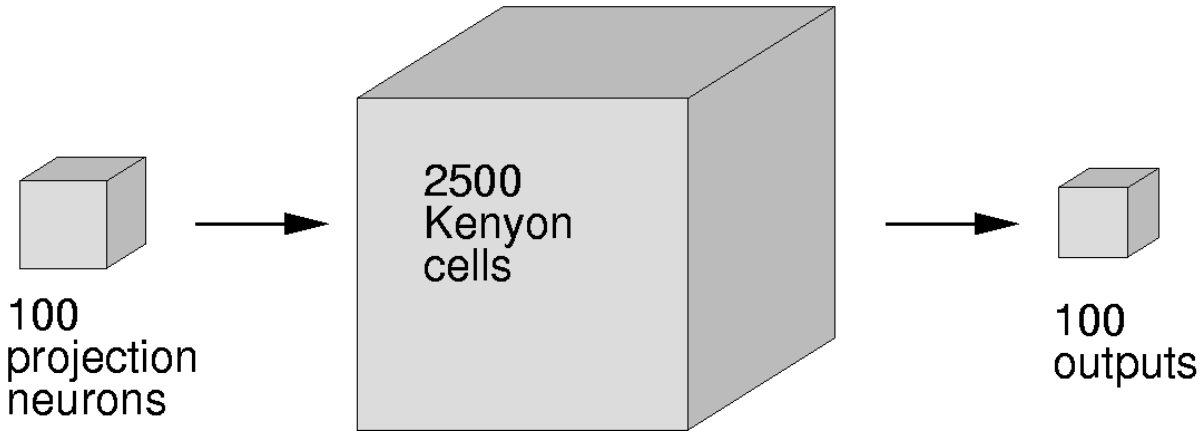
Perez-Orive et al., Science (2002)

"Have many, but only use a few"

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

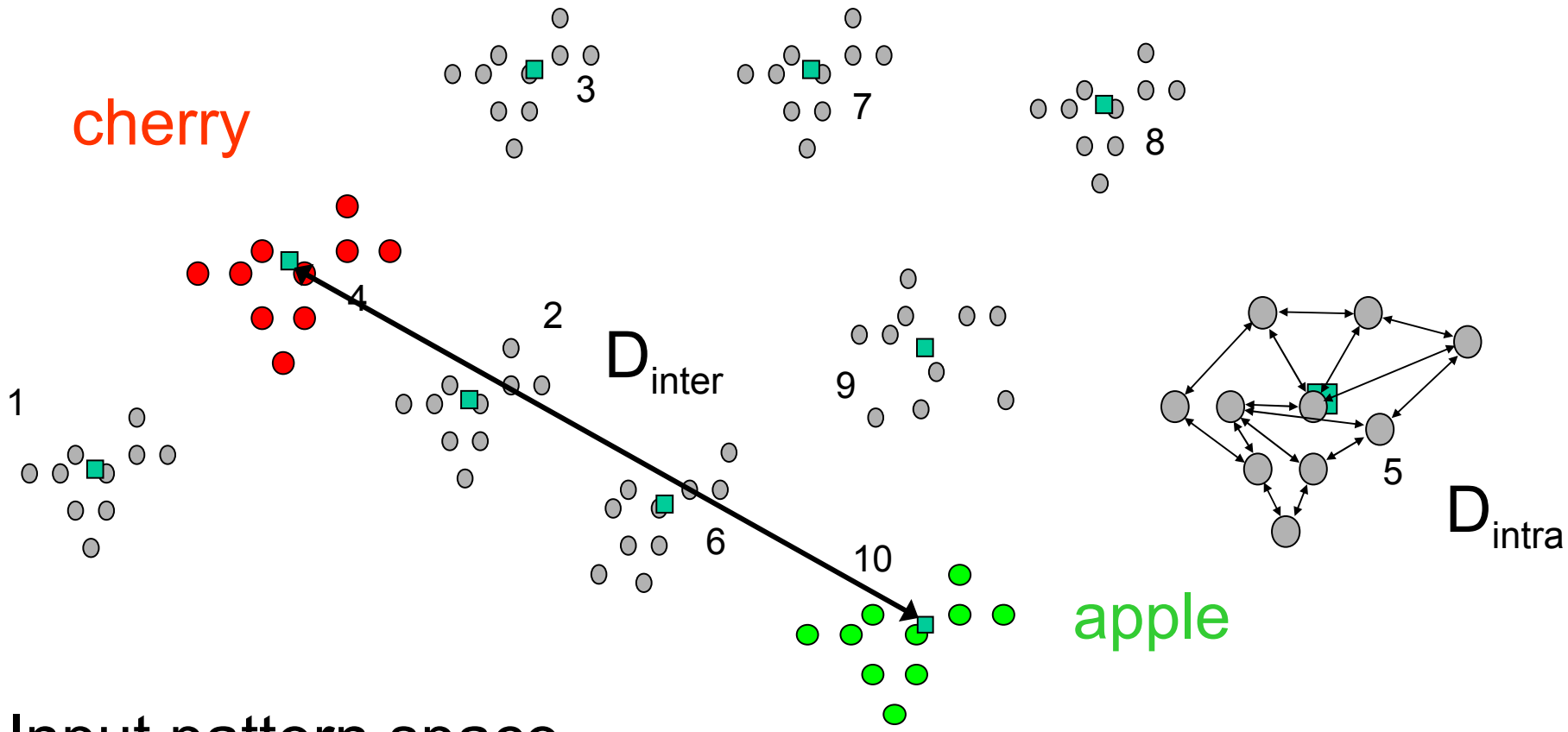
Classify classes of inputs

- 10 classes of inputs, 10 patterns each class
- “Winner-take-all” ouputs:
  The output neuron with the strongest input spikes
- Simulations in “*Drosophila* size”



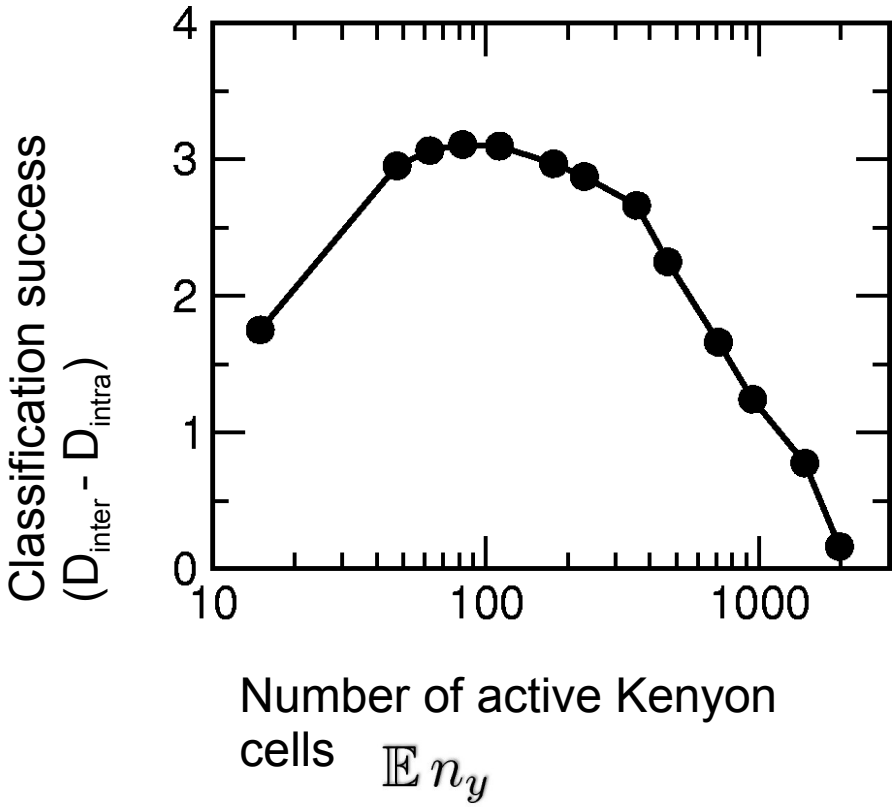
Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Classes of input patterns



Input pattern space

There are “optimal design parameters”



∃ Optimal $\mathbb{E}\, n_y$

of active Kenyon cells

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Summary: Connectionist model

- Random connectivity is enough for classification
- This suggests support vector machines with *random kernels* and *local*, “Hebbian” learning
- An optimal, sparse level of activity is postulated *and observed* in biology
- These systems are freely scalable & our analysis provides the parameters of choice
- These systems are extremely robust

Shortcomings

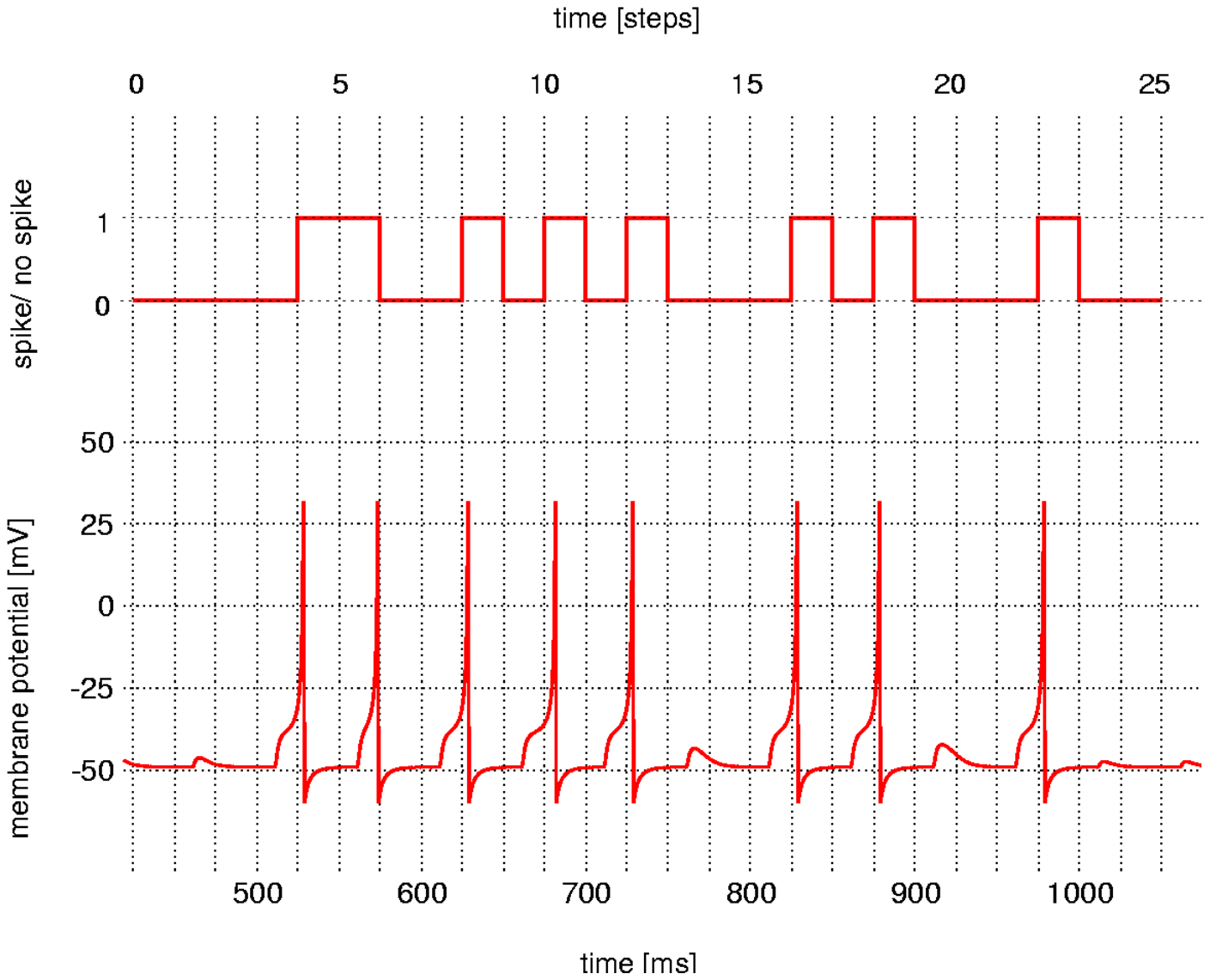
- The winner-take-all competition between output neurons has to be implemented artificially
- Gain control in the MB has to be implemented artificially

These issues can be resolved with more realistic spiking neuron models.

Spiking neuron models

McCulloch-Pitts

Spiking neurons



Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics

University of Sussex

“Rulkov model”

The figure on the last slide and the movies wer generated with the “Rulkov Model”:

The membrane potential is a discrete mapping from one time step to the next:

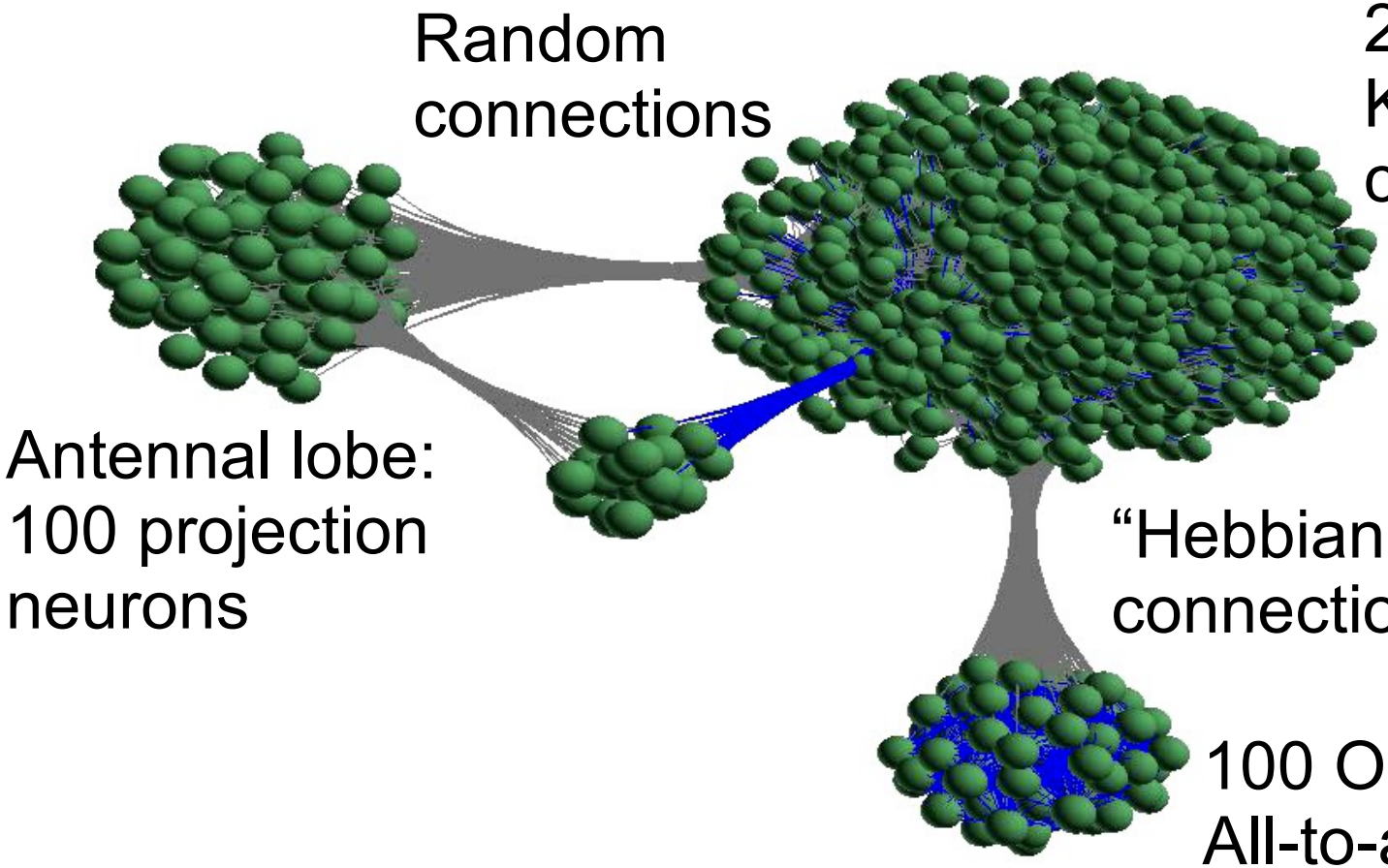
$$V(t+\Delta t) = \begin{cases} \dfrac{\alpha V_s^2}{V_s - V(t) - \beta I_{\text{syn}}} + V_s y & \text{if } V(t) \leq 0 \\ V_s(\alpha + y) & \text{if } (0 < V(t) < V_s(\alpha + y)) \\ & \quad \wedge (V(t - \Delta t) \leq 0) \\ -V_s & \text{otherwise} \end{cases}$$

We will go into more detail in the lab session.

Spiking neuron model

- Unlike in the previous models, we now implement competition in the MB lobes by all-to-all inhibitory synapses
- Learning is now entirely unsupervised
- The system does not even know that there are classes and how many there are.

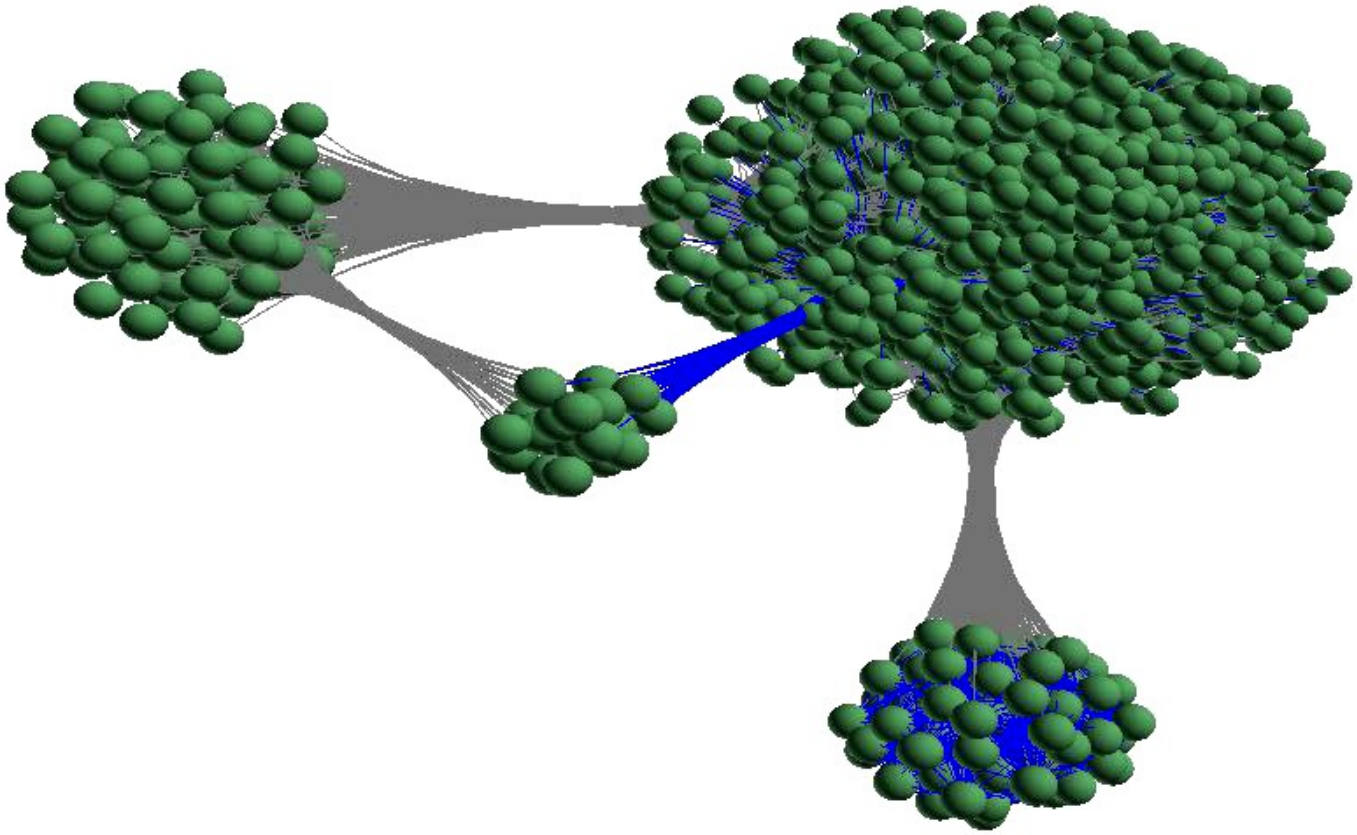
Process of recognition: Naïve locust



Created with neuranim
http://sourceforge.net/projects/neuranim

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Experienced locust



Created with neuranim
http://sourceforge.net/projects/neuranim

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Quantification

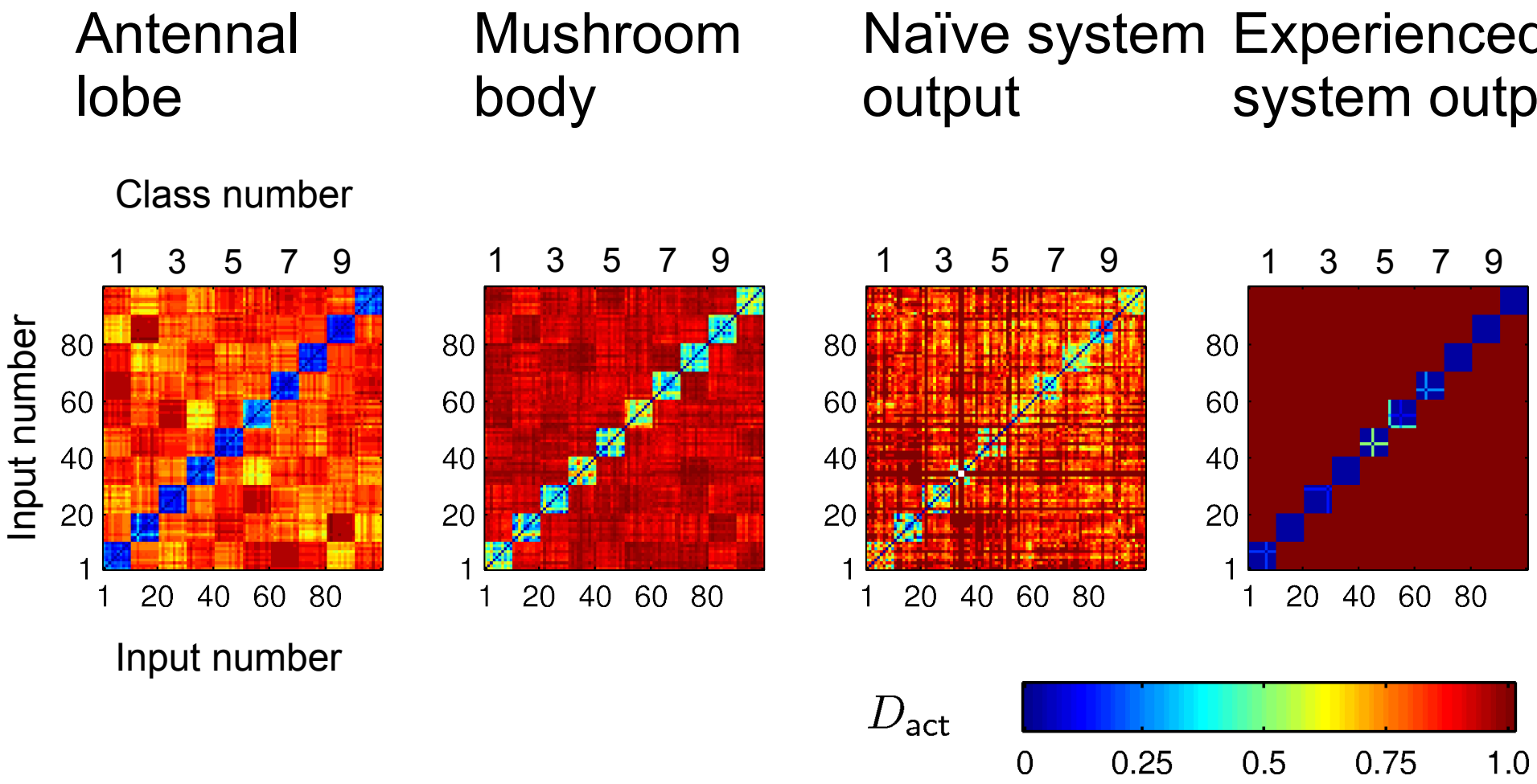
Pairwise Humming distance of patterns normalized by activity:

$$|\mathbf{z}| = \sum_{k=1}^{N} |z_k|$$

$$D_{\mathrm{act}}(\mathbf{z}^{(i)}, \mathbf{z}^{(j)}) = \frac{|\mathbf{z}^{(i)} - \mathbf{z}^{(j)}|}{|\mathbf{z}^{(i)}| + |\mathbf{z}^{(j)}|}$$

Note: D_{act} is between 0 and 1.

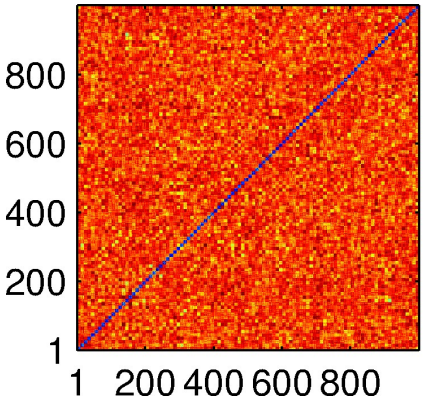
Quantitative Analysis



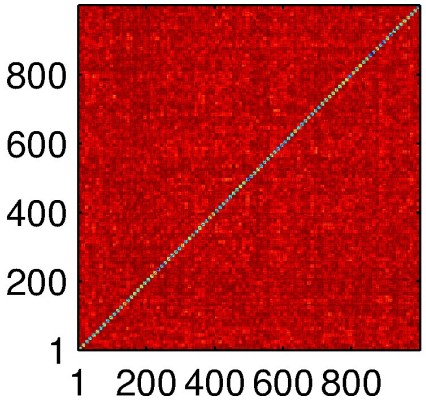
Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Quantitative Analysis

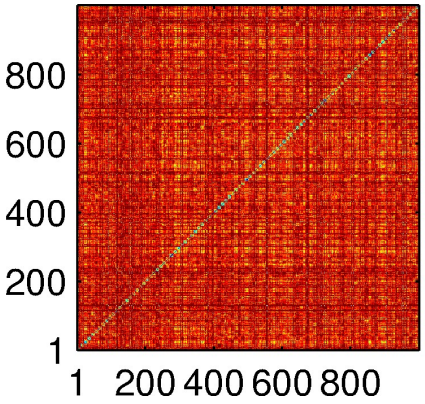
Antennal lobe



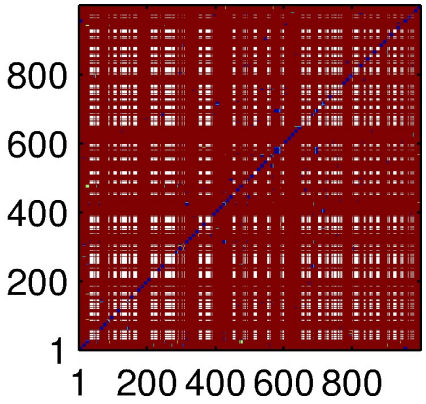
Mushroom body

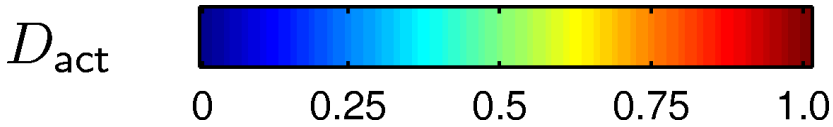


Naïve system output



Experienced system output





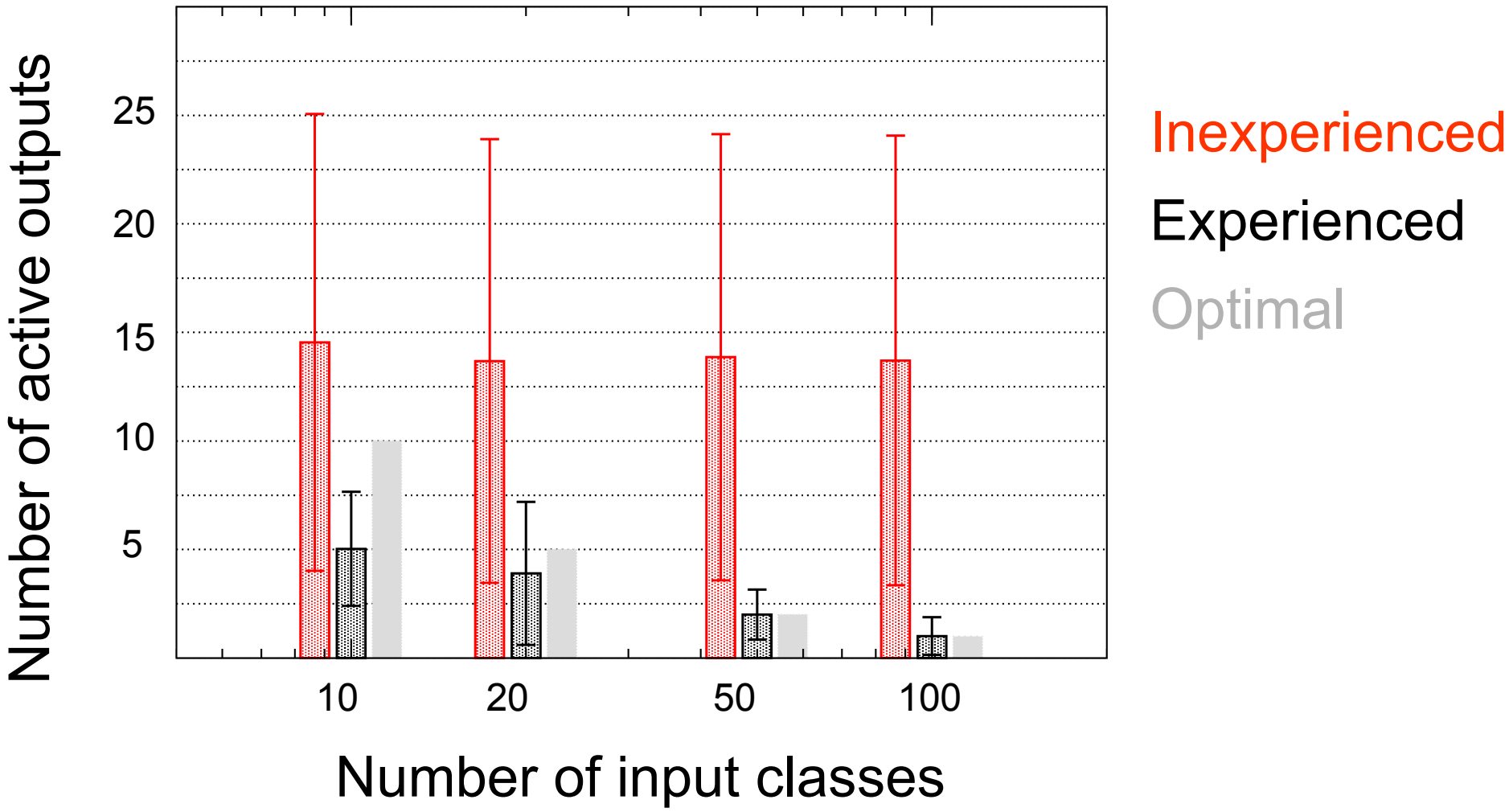
Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics

University of Sussex

Self-organization of output responses

- In principle, all or any of the output neurons can respond at any given time
- Which do, and which don't depends on the competition between them.
- *A priori it is unclear how many will respond*

Automatic detection of input set structure



Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Summary

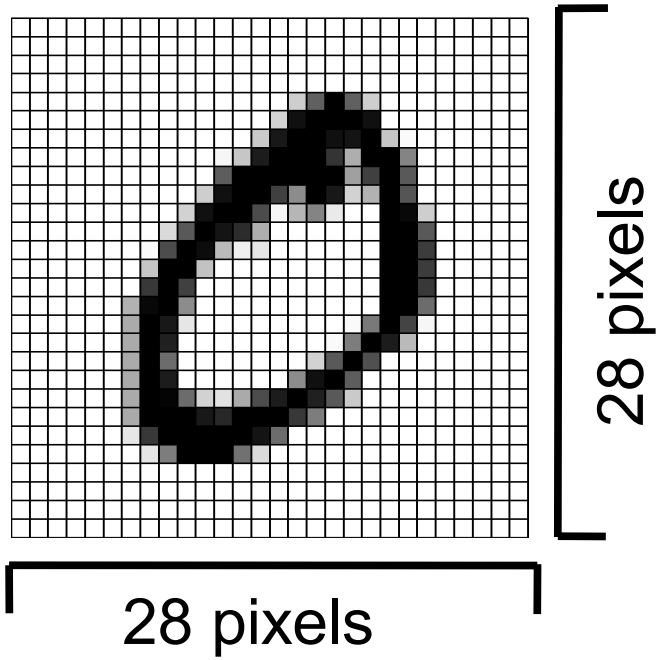
- More realistic biophysical models demonstrate that the system can *self-organize* to recognize odors
- The system detects *the structure of the input pattern set* autonomously

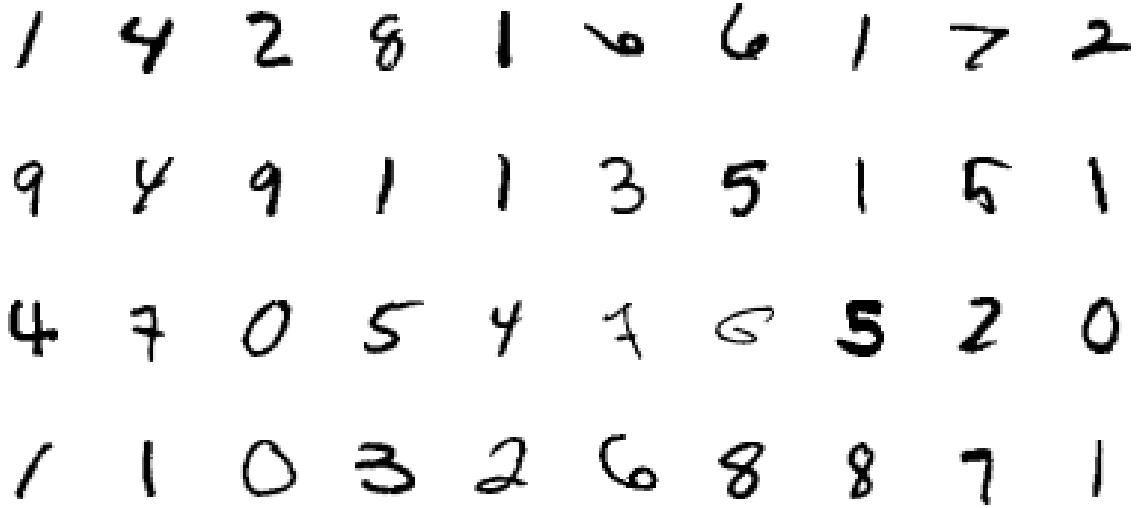
Benchmarking

- To address criticism that synthetic data is hard to judge we benchmarked against a standard pattern recognition problem:
- MNIST data set of handwritten digits

Benchmark: MNIST database

256 level grayscale

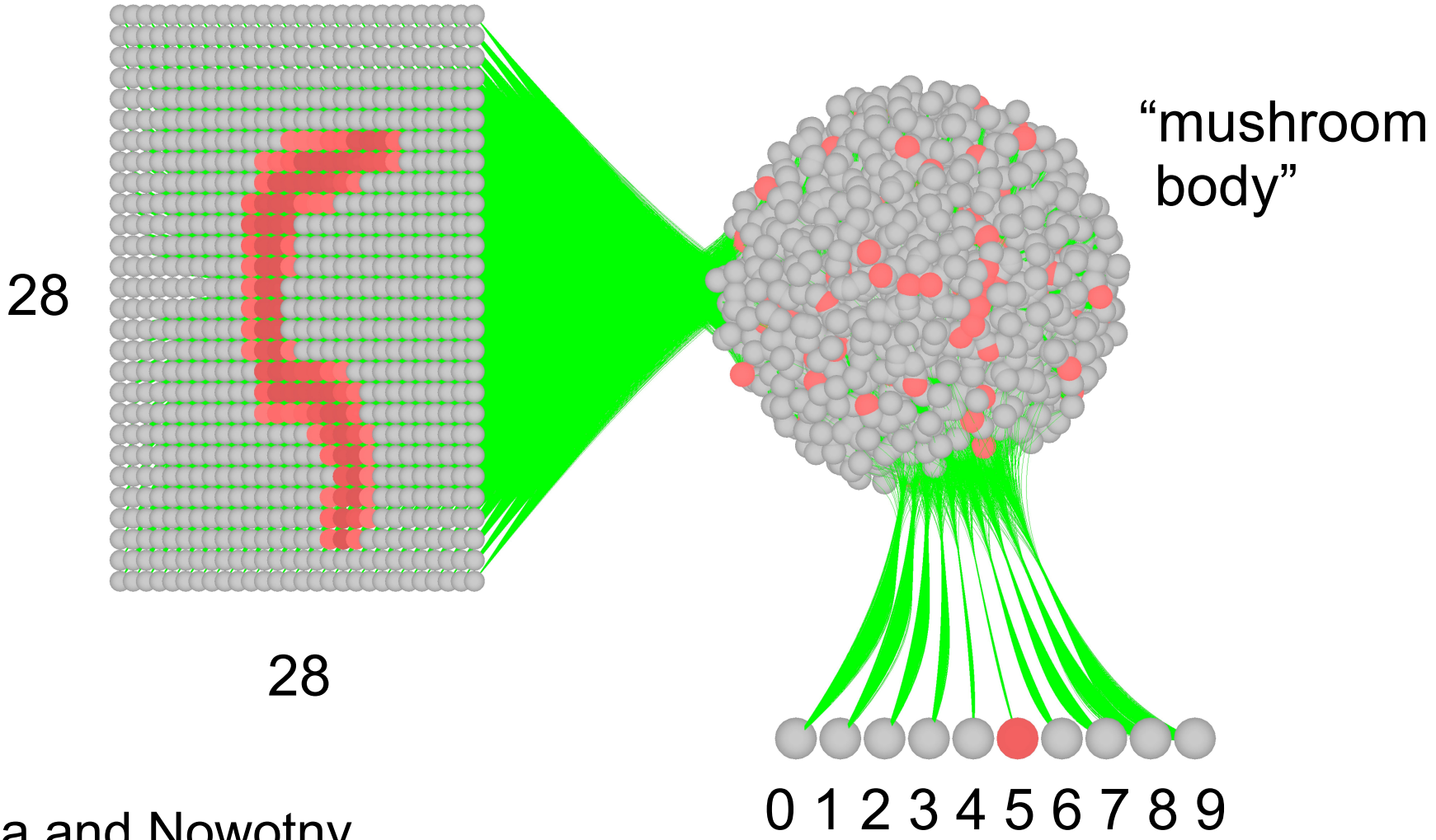




- Centering (x and y) & size normalisation
- 60000 – training
- 10000 - testing

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Using the ‘locust olfactory brain for digit recognition



Huerta and Nowotny,
Neural Computation (2009)

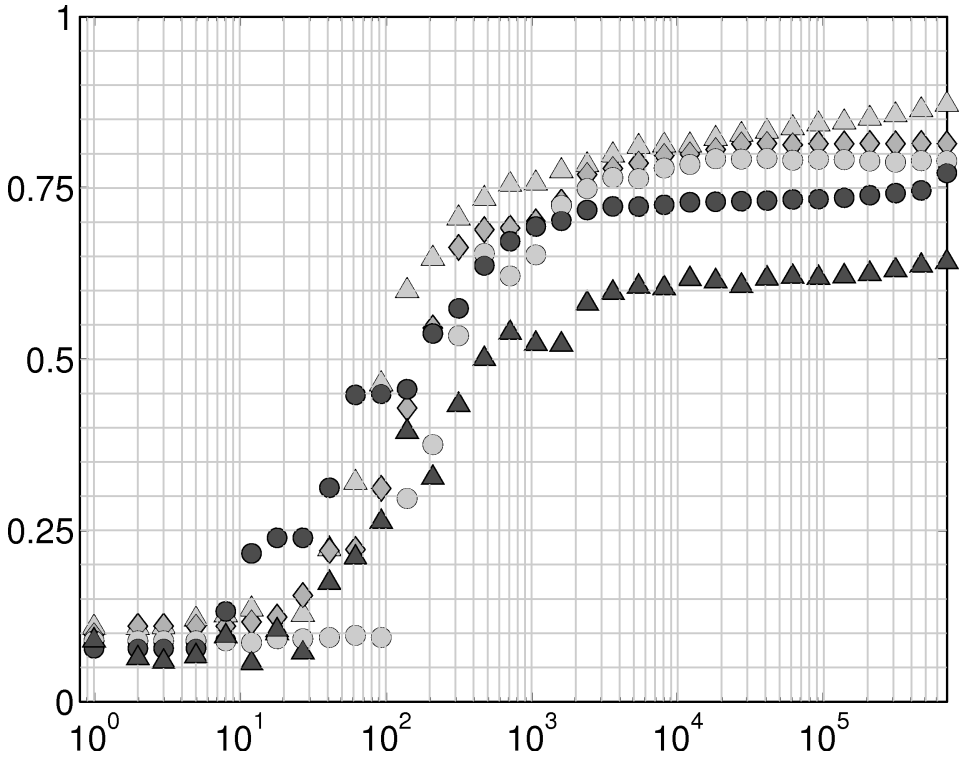
Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Note ...

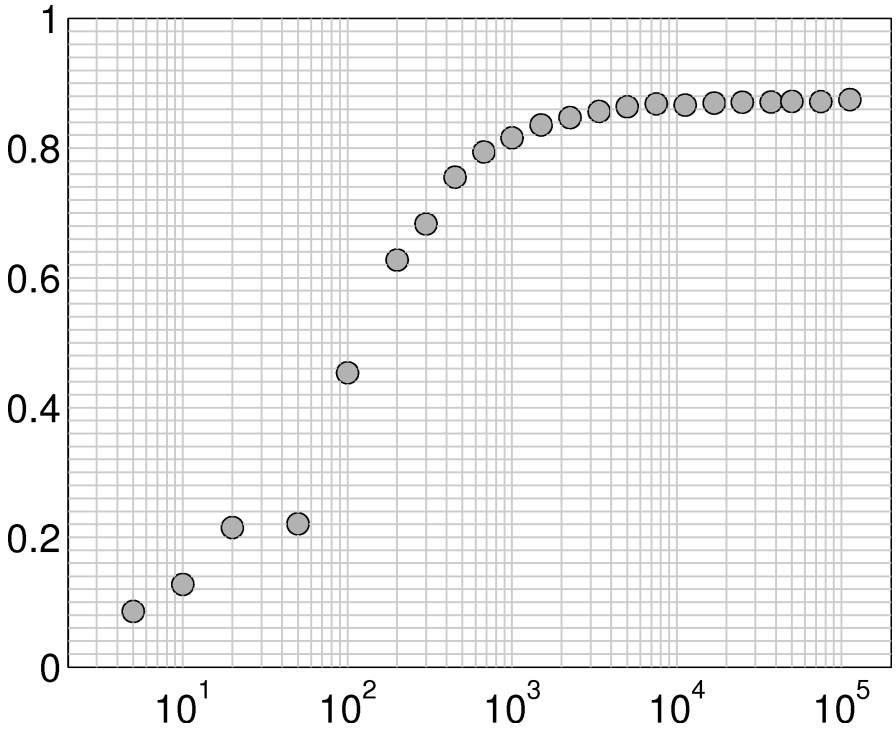
This is actually not as strange as it may seem at first glance:

The mushroom bodies have been implicated in vision as well as in olfaction, they are likely a multi-modal “learning centre”

Classification performance



Number of training samples



Number of Kenyon Cells

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics

University of Sussex

Last step: hardware acceleration

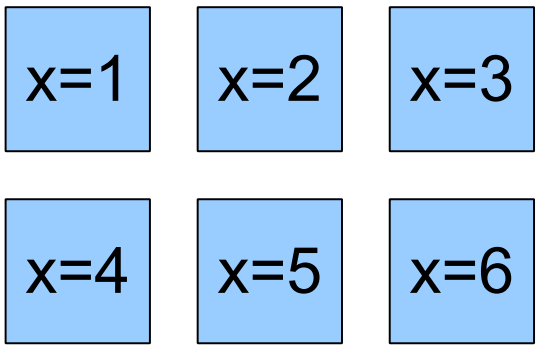
The model is numerically demanding (on the order of 60000 cells and a million synapses).

If we want to use it for applications we will need to parallelize its operation to optimize the speed.

We have built a prototype on a GPU in the NVidia CUDA framework.

NVidia CUDA

- “Common Unified Device Architecture”
- Allows main stream developers to use massively parallel graphics chips for general purpose computing



Code split into “kernels”
A set of kernels form a “grid”

Grids are executed partially in parallel, partially in series

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Lastly: Implementation of the system on NVidia CUDA

TESLA S1070 GPU
256 thread processors

1.5 GHz

Kenyon Cell Kernel

- Download incoming c
- 1000x
  * Download inputs
  * calculate 1000 ouputs
  * write back outputs

512 parallel invocations, 50000 total

Output Neuron Kernel

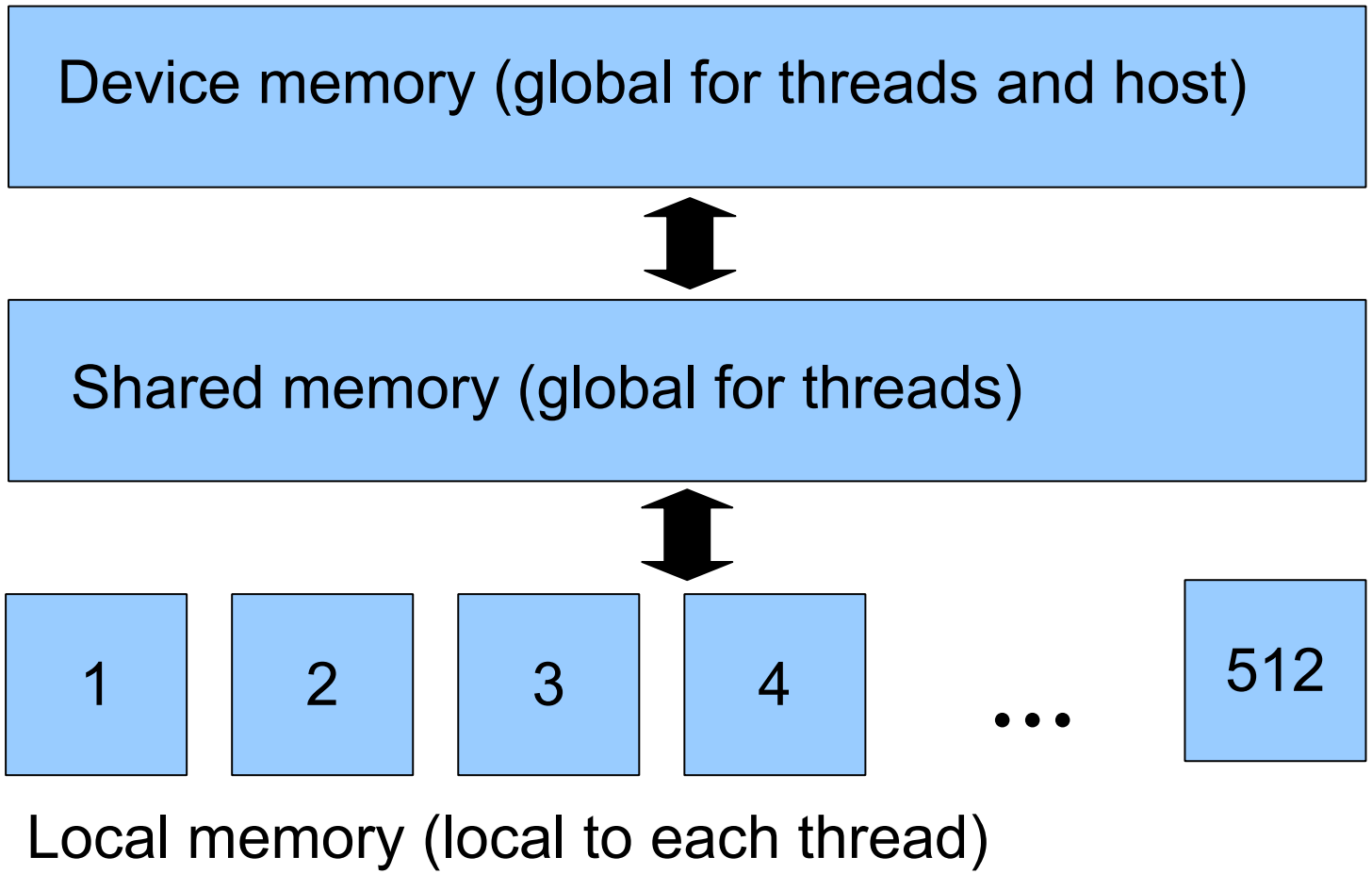
- Download KC patterns
- 1000x
  * Calculate output
  * write back output
  * adjust synapses

500 parallel invocations

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics

University of Sussex

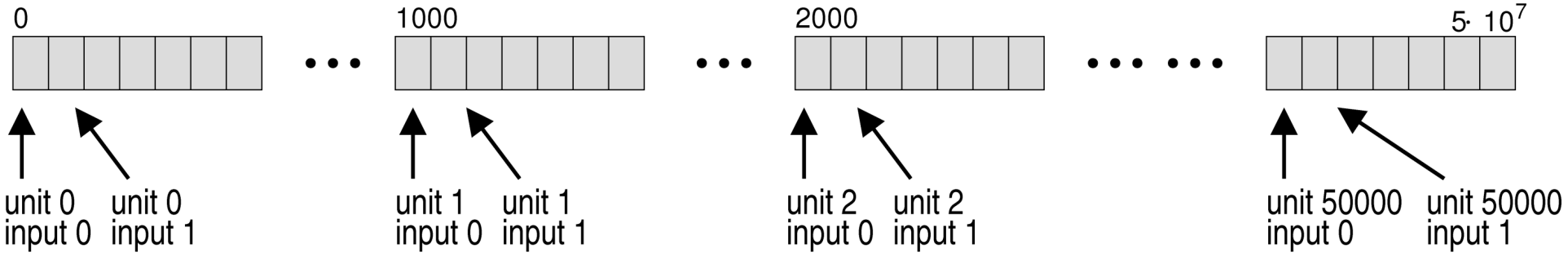
CUDA memory hierarchy



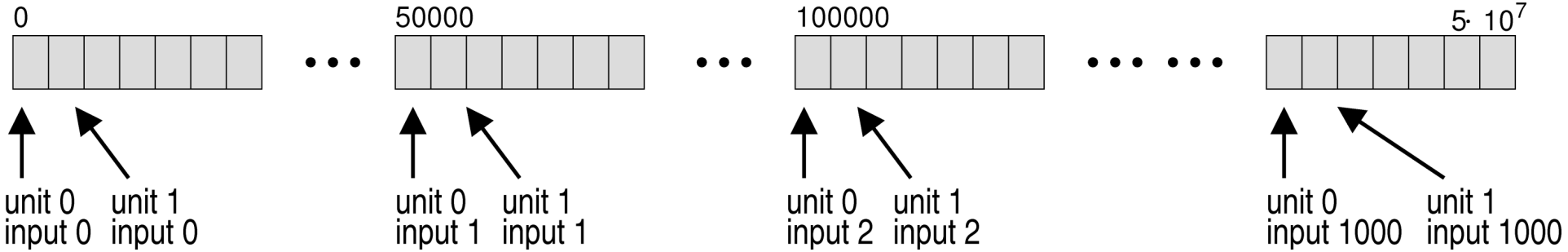
Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Memory usage models

Device memory ordered by input locally and unit globally

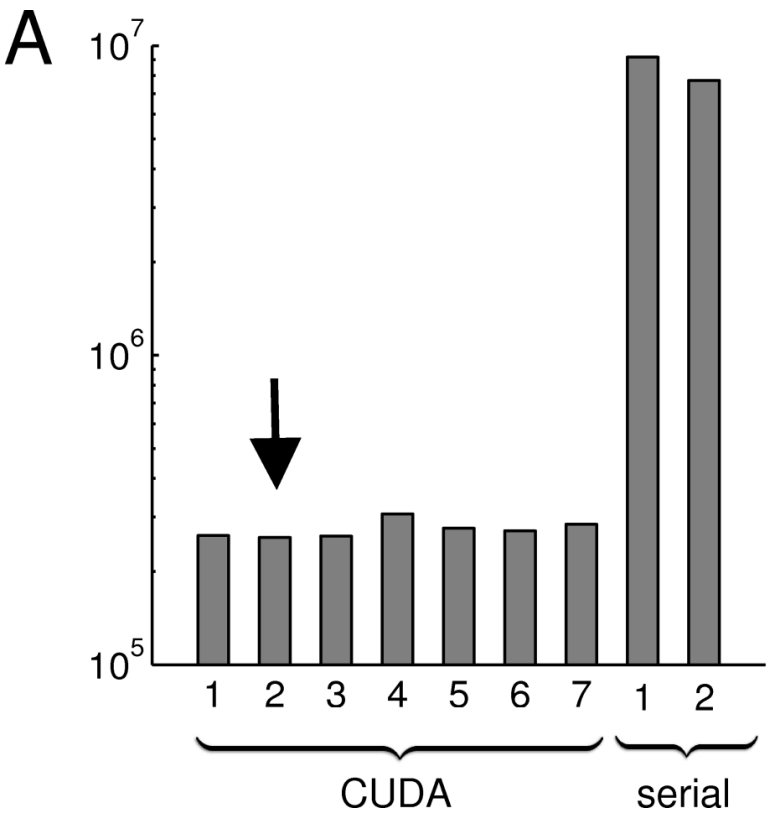


Device memory ordered by unit locally and input globally

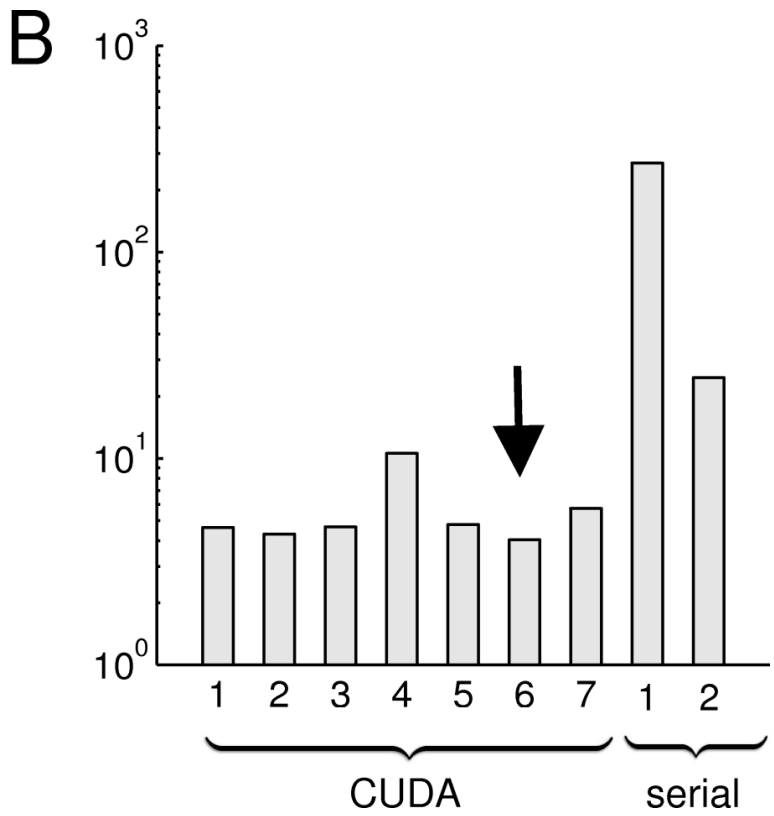


Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Summary of Timing results



Constant time requirement (loading digits, connectivity, preprocessing, testing)



Time requirement proportional to number of digits trained

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics

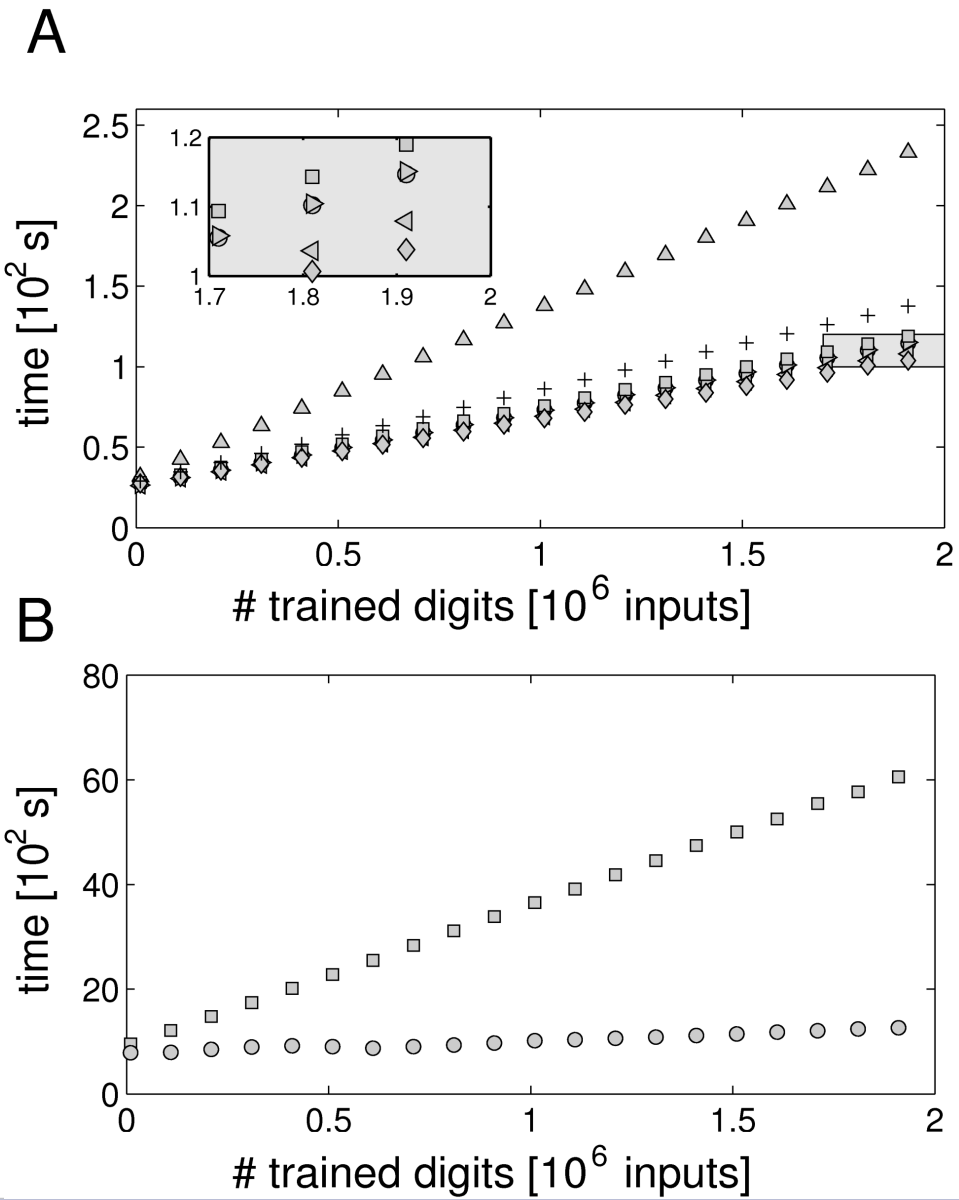
University of Sussex

7 CUDA implementations

- KC as bits in array of bytes; input number locally
- Every 8th thread sets bits in a local byte-size buffer, copies this to device memory; unit number

  locally
- As 1 but 32 bit integers
- As 2 but 32 bit integers
- KC as byte-sized integers directly to the device memory; input number locally
- As 5 but reverse ordering scheme
- As 5 but 32 bit integers

Timing trials



CUDA

Worst: Assembling 32bit locally

Best: Direct bytes to device; unit locally

Host

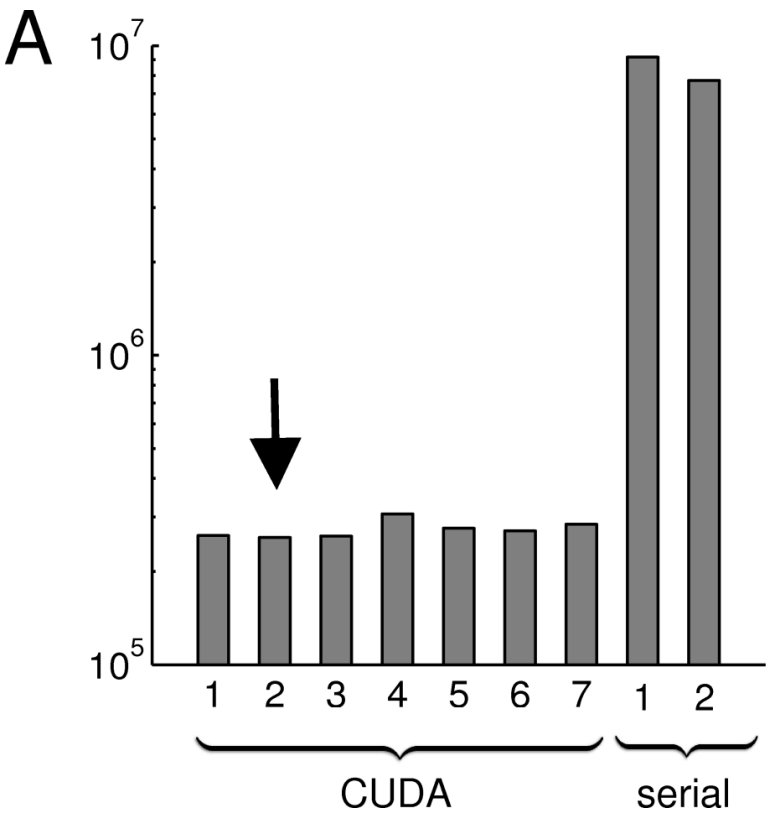
Good: Unit local

Bad: Input local

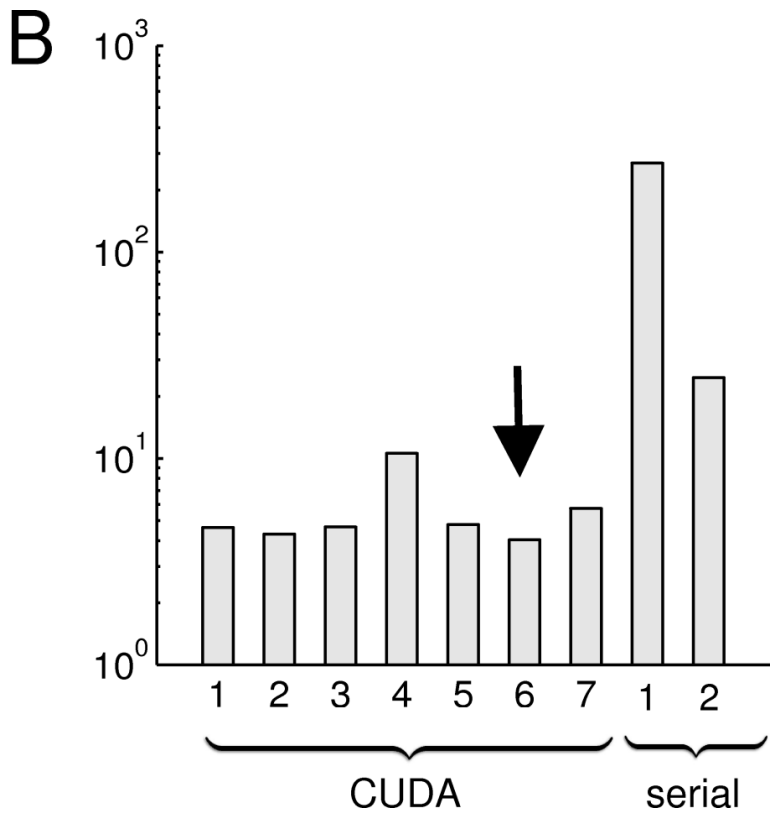
Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics

University of Sussex

Summary of Timing results



Constant time requirement (loading digits, connectivity, preprocessing, testing)



Time requirement proportional to number of digits trained

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics

University of Sussex

Discussion: CUDA implementation

- 30 fold speed increase for KC evaluation (hidden layer)
- 6 fold speed increase overall
- Optimization of memory access is extremely important
- Things will become truly interesting when a full classifier can be done in one kernel grid invocation (that fits onto the device)

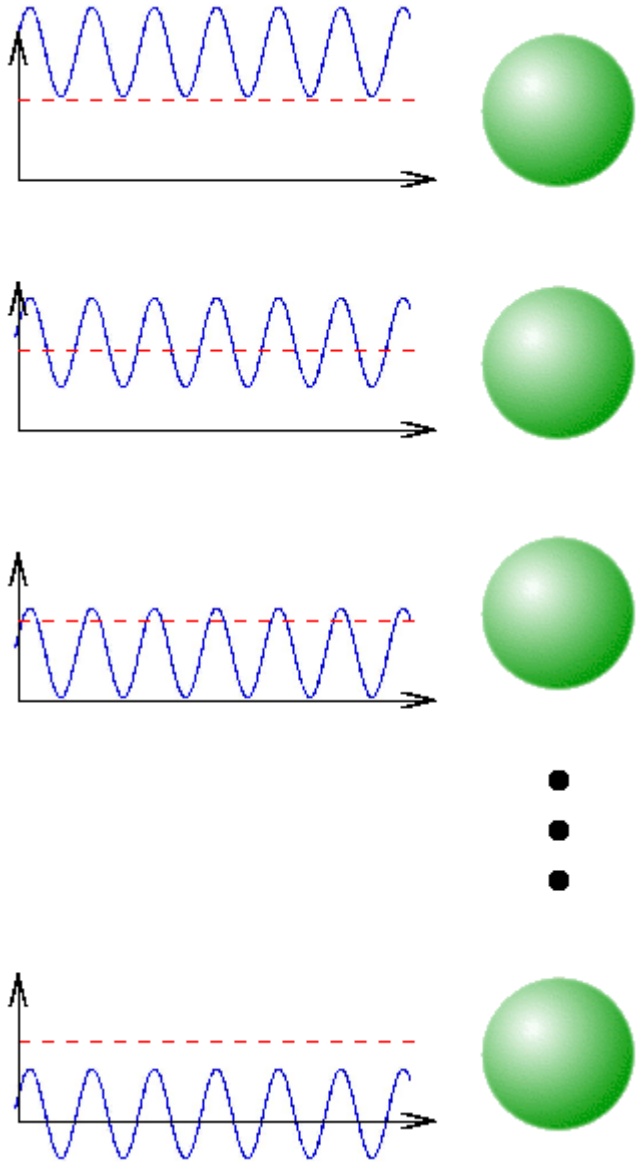
Brody & Hopfield

Model of olfactory processing

Hopfield's model of olfaction

- This is not **the** Hopfield model
- This model is based on what Brody and Hopfield call “Many Are Equal”
- This is based on a fundamental mechanism of synchronization by sub-threshold oscillations

Synchronziation by sub-threshold oscillations



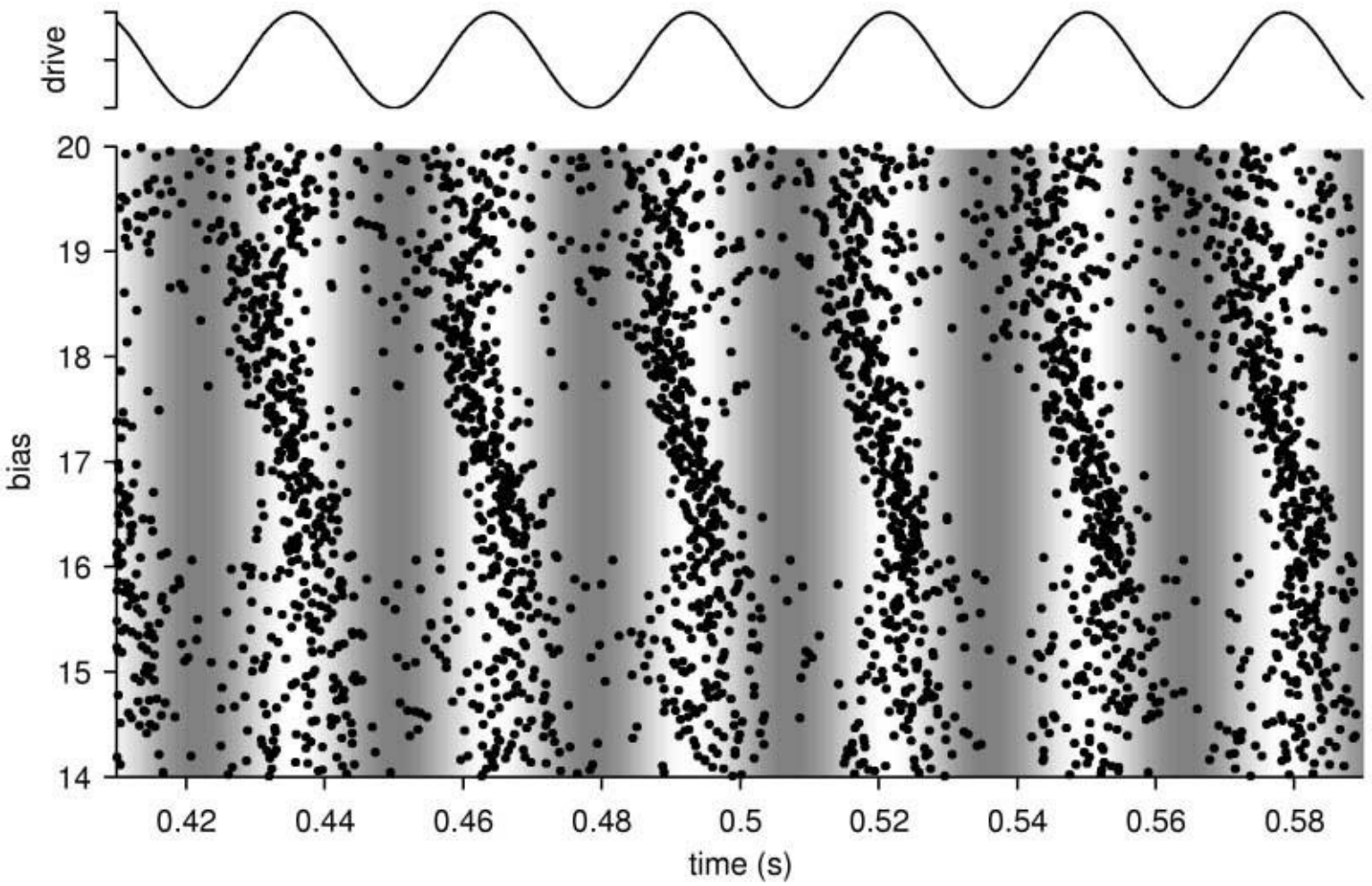
$$I_1 = I_{\text{offset},1} + I \sin \omega t + \text{noise}$$

$$I_2 = I_{\text{offset},2} + I \sin \omega t + \text{noise}$$

$$I_n = I_{\text{offset},n} + I \sin \omega t + \text{noise}$$

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Synchronization by sub-thresold oscillations



Brody & Hopfield, Simple Networks for Spike-Timing-Based Computation, with Application to Olfactory Processing, Neuron **37:** 843-852 (2003)

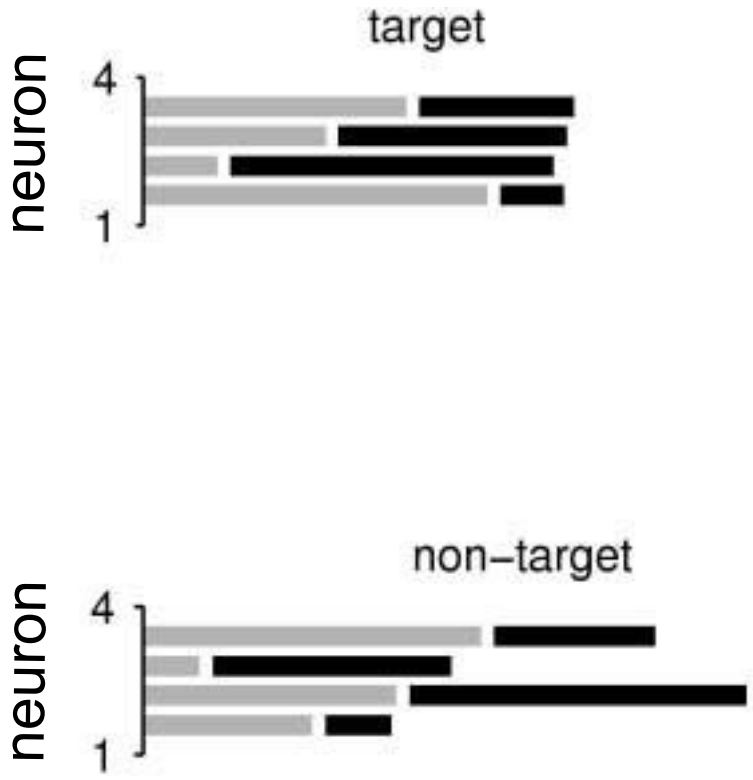
Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics

University of Sussex

Recognition by coincidence detection

- This implies that neurons that receive the same constant input current fire at the same time
- Coincidence of spikes implies identical input.

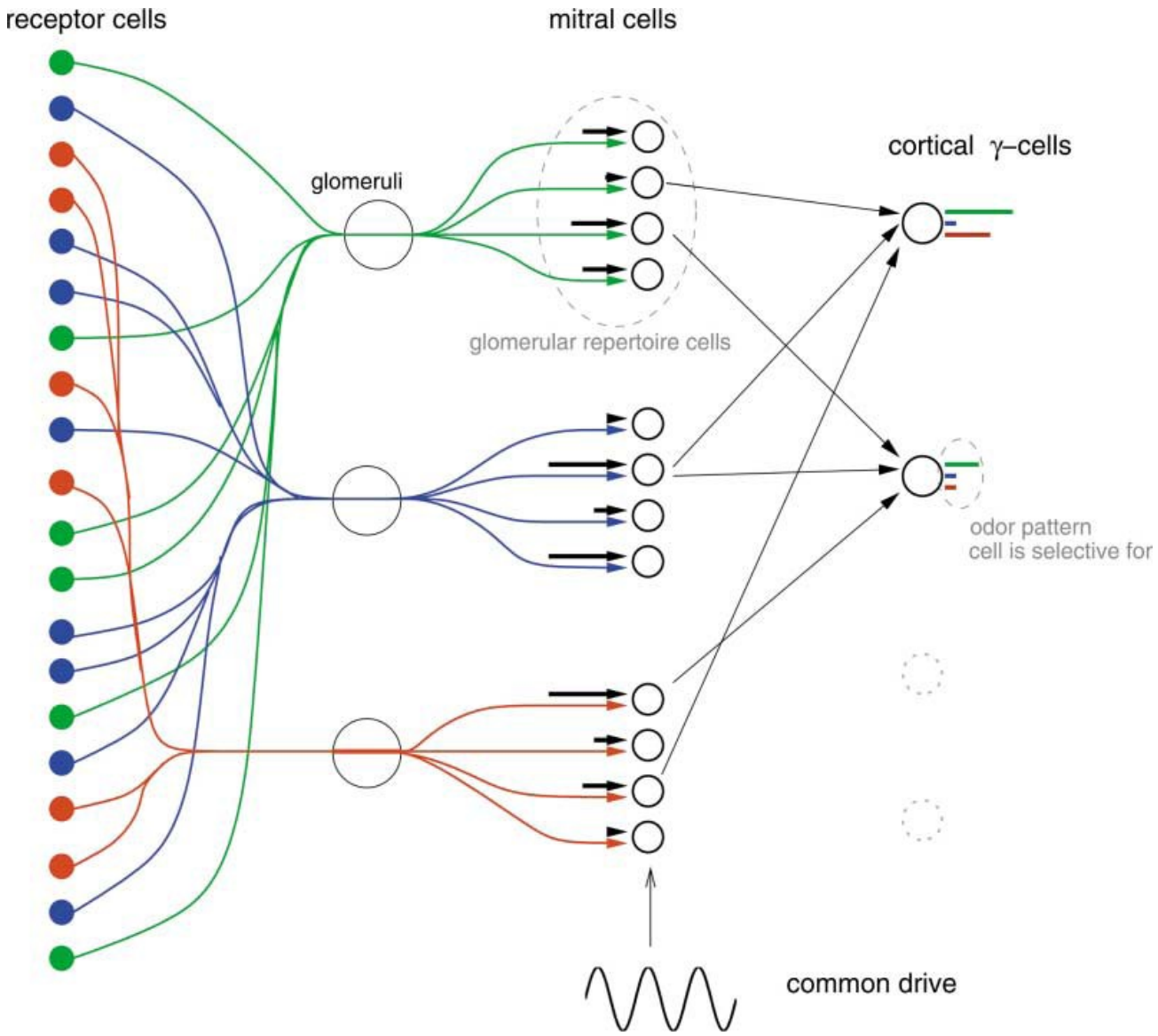
Key – lock principle



- Grey – constant bias current in each “mitral cell”
- Black – input current evoked by an odor input
- If the input “is right”, all neurons receive the same input current and thus spike synchronously

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Hopfield's olfaction model

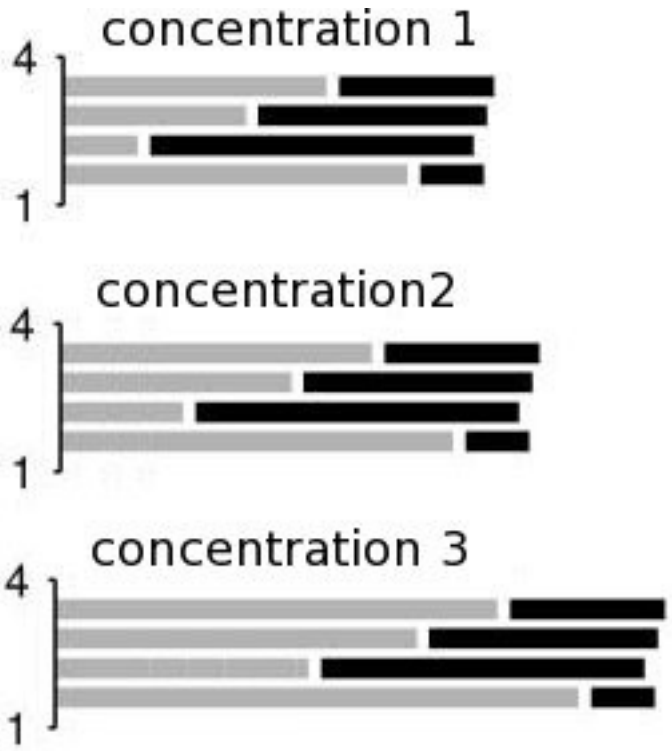


- The cortical cells connect to the mitral cell with the “correct bias”
- Odors are detected when the cortical cell gets synchronized input
- 400 ORN types, each odor excites 200

Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Discussion

- Odors are recognized reliably across a large range of concentrations



Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Discussion

- Odors are recognized against a stronger background odor
- Odors in a mixture can be recognized separately (if the set of active glomeruli does not have too much overlap)
- Odors in a binary mixture with fully overlapping glomerulus set can sometimes be recognized as well (?)

You can look at these points more with the Exercises.