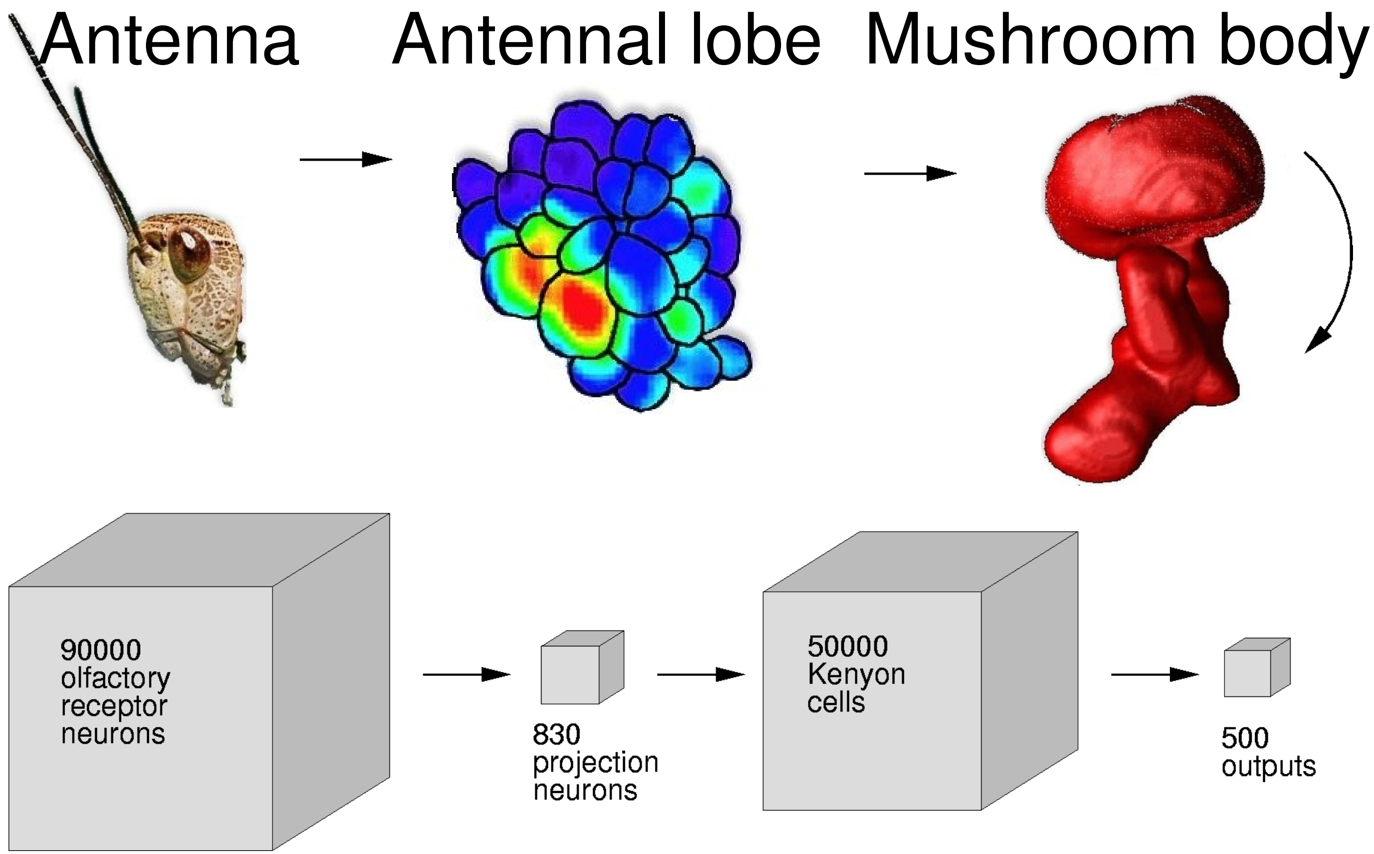
Lecture 3:
Modelling Insect Olfaction

Dr. Thomas Nowotny
University of Sussex

Why olfaction of insects?

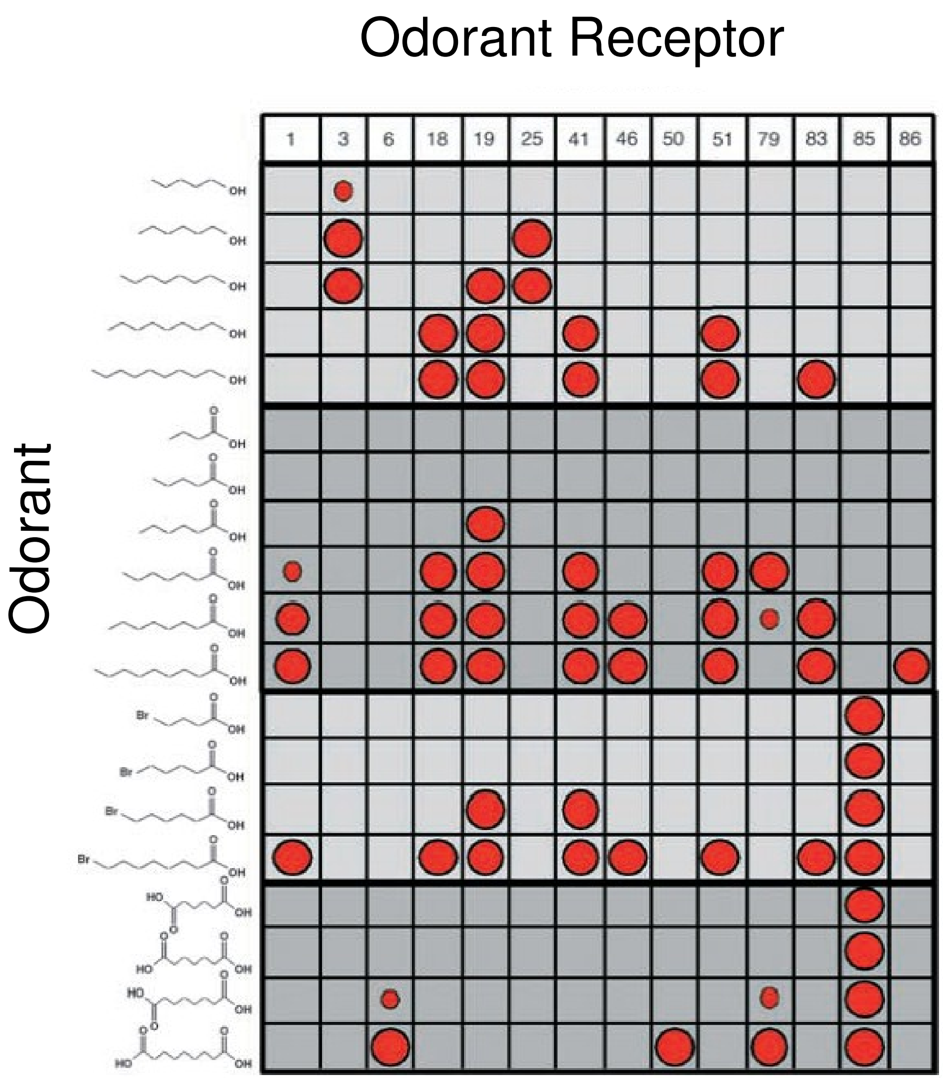
- Biological sensory systems have an amazing performance
- Insect olfaction is a good model to understand sensory processing
  - The systems are comparably small and experimentally accessible
  - Structure is very similar across species
  - Many recent advances (Nobel prize 2004)

Main olfactory pathway anatomy



Box volume ~ number of cells

Olfactory Receptors



Odors evoke different, but overlapping patterns of receptor activity

From Linda Buck: Nobel lecture

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

Odors are encoded as overlapping patterns of projection neuron activity.

In Richard Axel's words

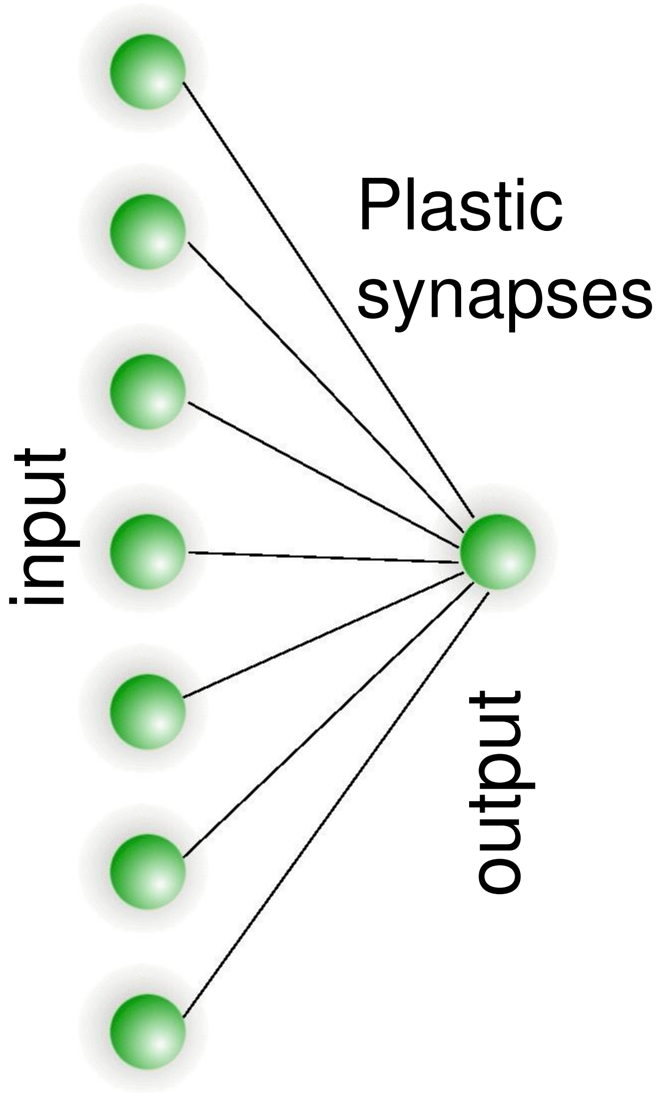
“The elucidation of an olfactory map […] leaves us with a different order of problems. Though we may look at these odor-evoked images with our brains and recognize a spatial pattern as unique and can readily associate the pattern with a particular stimulus, the brain does not have eyes. “

Richard Axel, Nobel lecture

In other words:

The algorithm of olfactory information processing remains to be found.

A classical pattern recognition solution

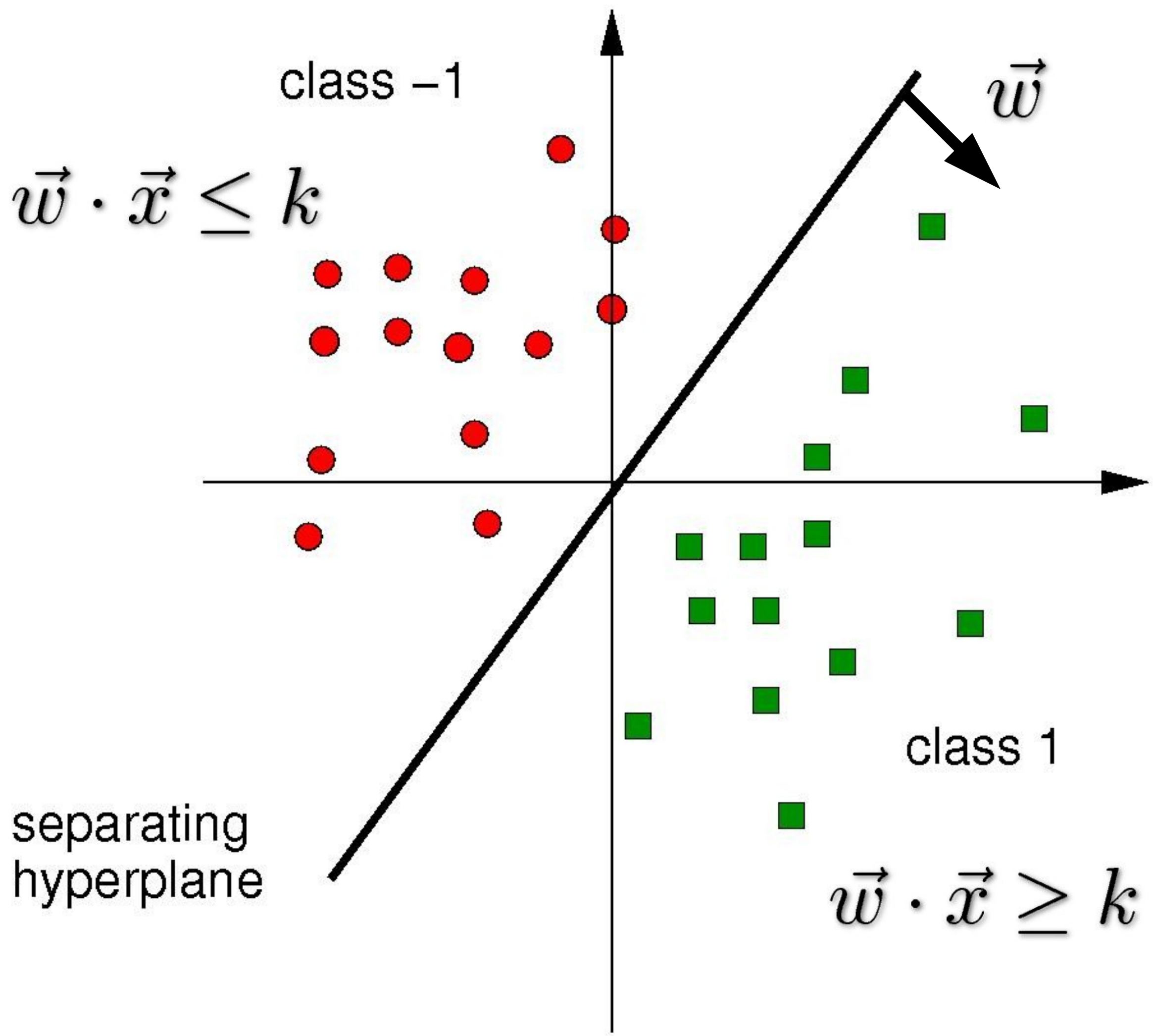


A simple perceptron rule:

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)

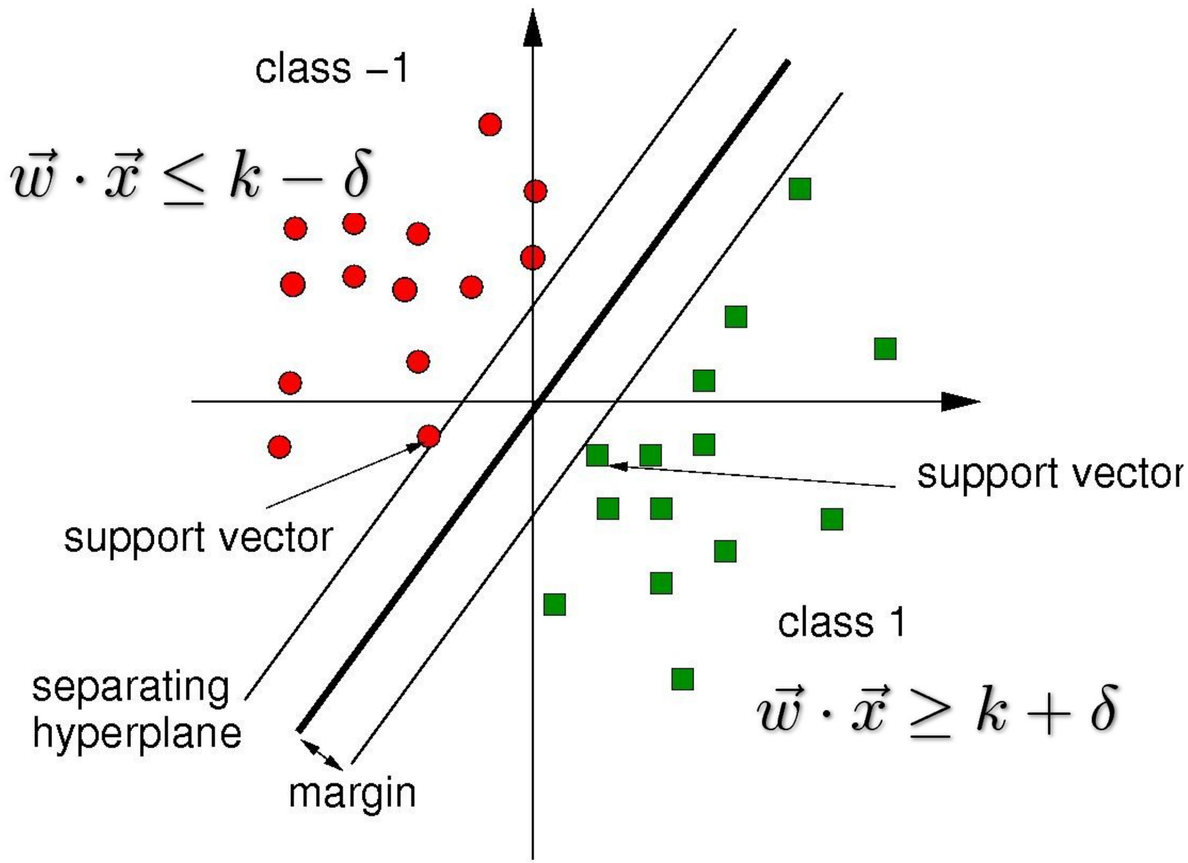
The perceptron is a linear classifier



The hyperplane is adjusted through the training and Hebbian learning

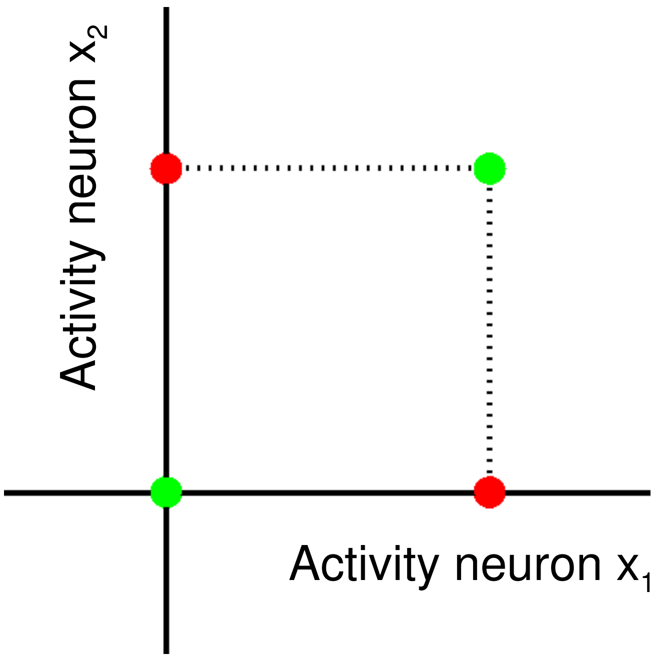
Support Vector Machines (SVM)

Cortes and Vapnik 1992,95: Support vector machine:



Here the hyperplane is adjusted to maximise the margin

Linear Classification can fail



There is no line that can separate green from red.

Dimension = number of neurons

Thomas Cover, 1965

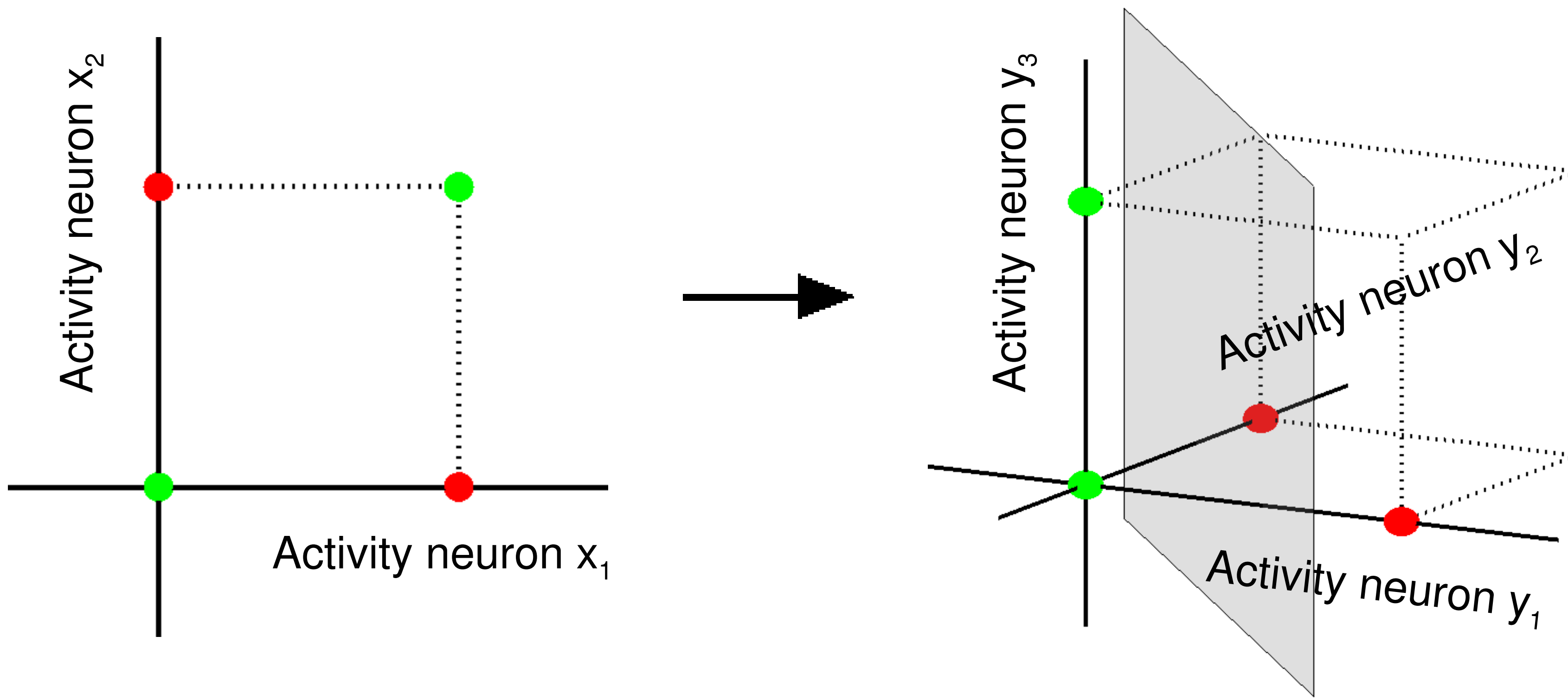
"Classification is much more probable if the input is first cast into a high-dimensional space by a non-linear transformation."

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

Kernel trick



Dimension = number of neurons

Typical kernels (transformations) used

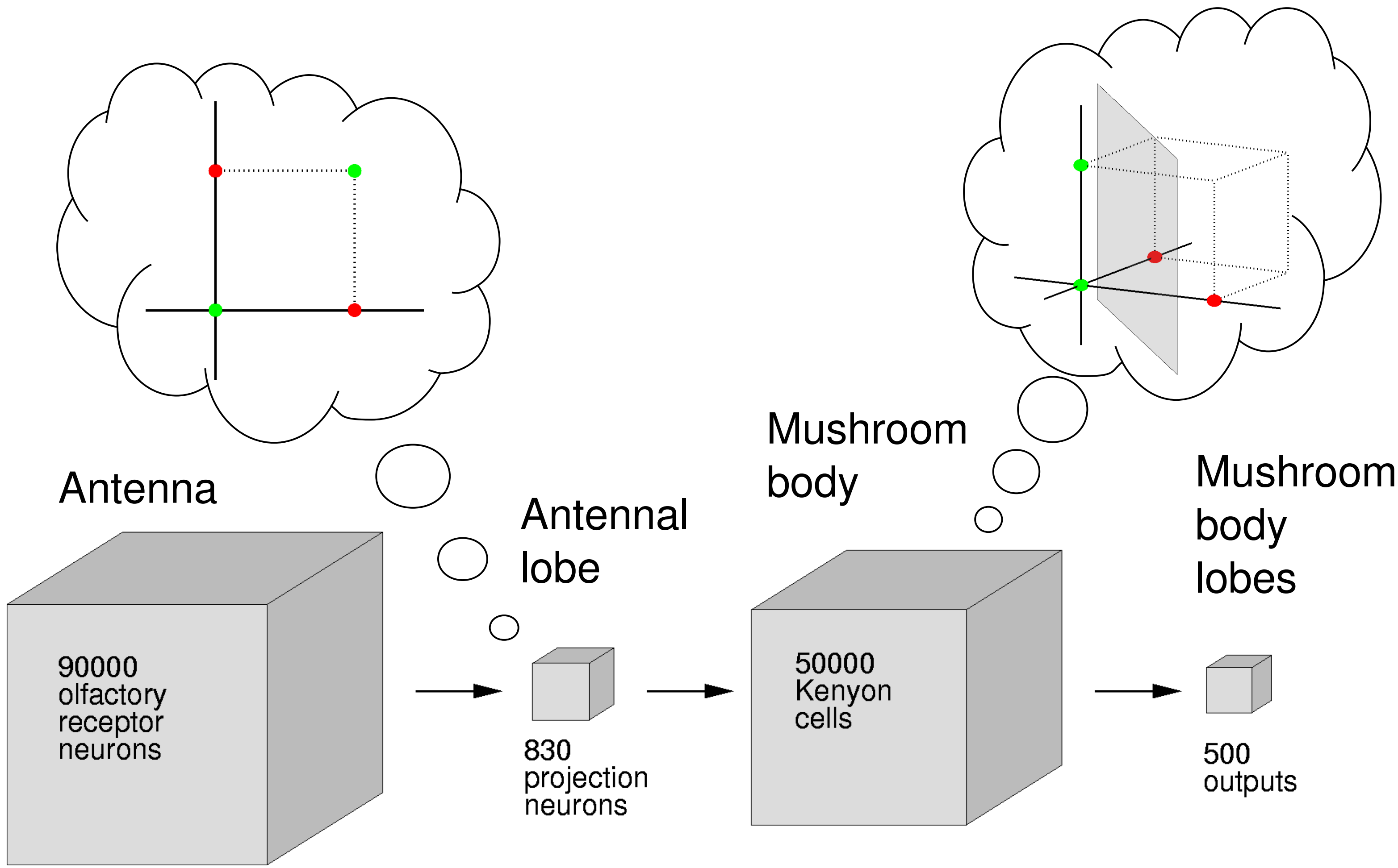
Polynomial (homogeneous): $K(\vec{w},\vec{x}) = (\vec{w}\cdot\vec{x})^j$

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

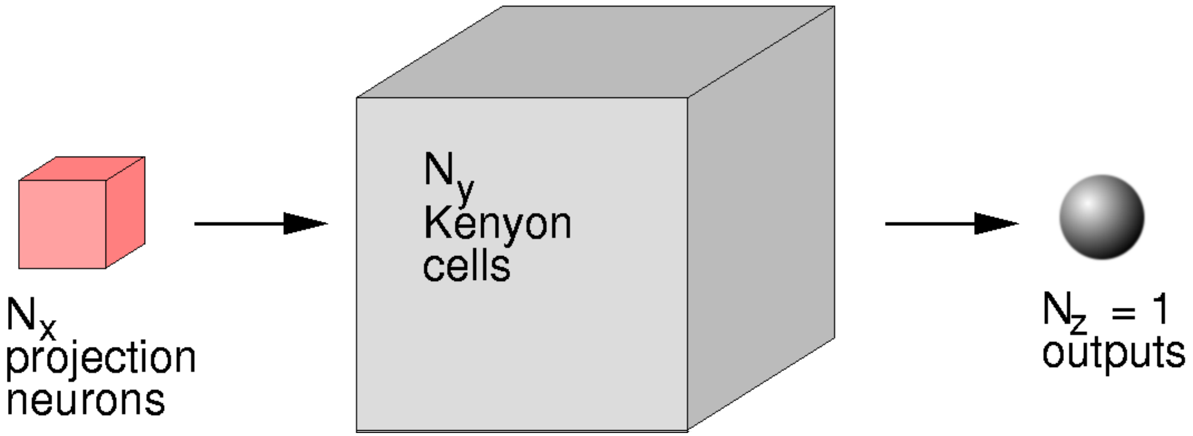


But we will use a *random kernel* (random connections)

Classify one pattern from the rest

Random input patterns

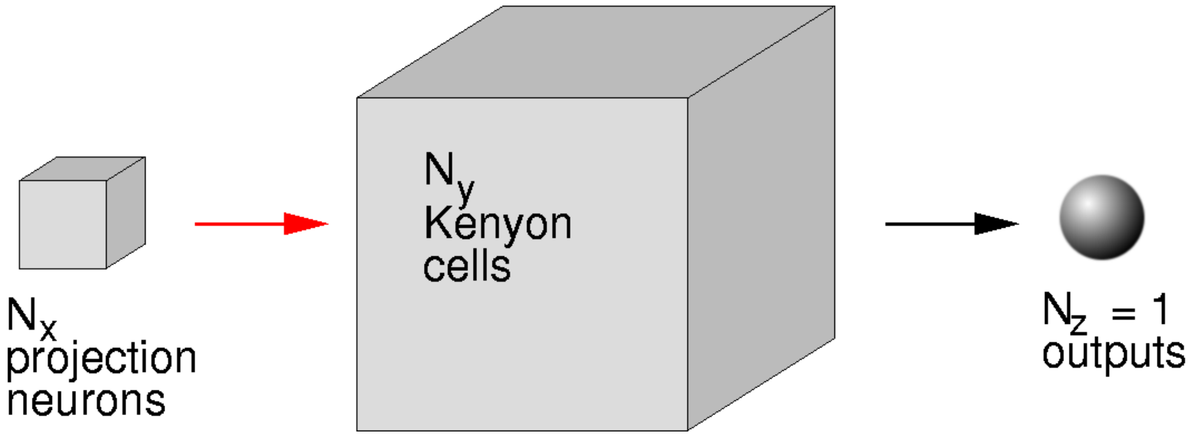
$$x_i = \begin{cases} 1 & \text{with} \quad p_x \\ 0 & \text{otherwise} \end{cases}$$



Classify one pattern from the rest

Random connections

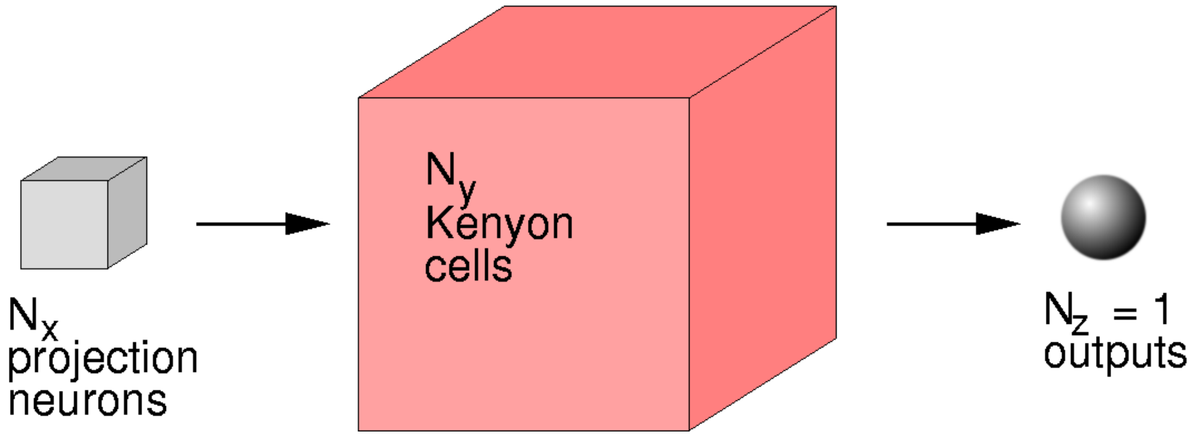
$$w_{ji} = \begin{cases} 1 & \text{with} \quad p_{y \leftarrow x} \\ 0 & \text{otherwise} \end{cases}$$



Classify one pattern from the rest

McCulloch-Pitts neurons

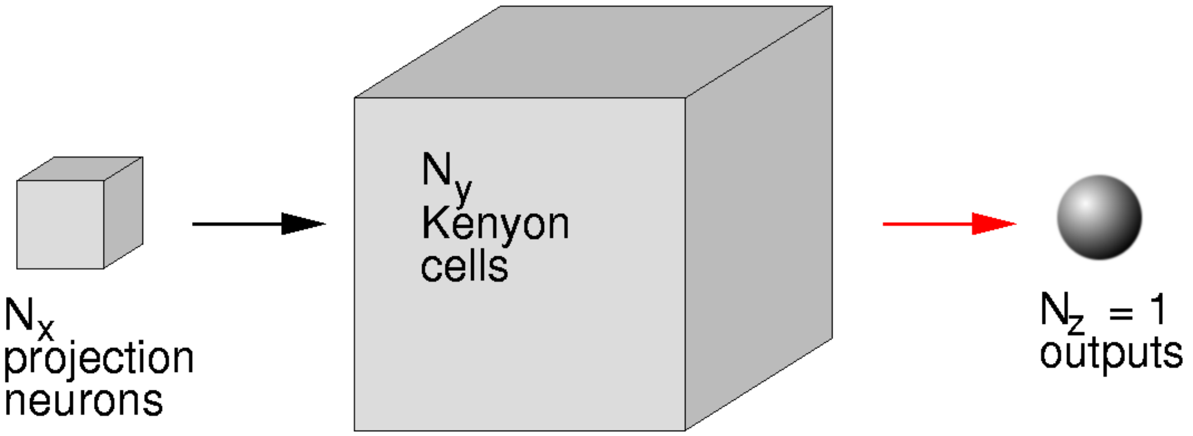
$$y_j(t) = \Theta\left(\sum_i w_{ji} x_i(t-1) - \theta \right)$$



Classify one pattern from the rest

"Hebbian" connections

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

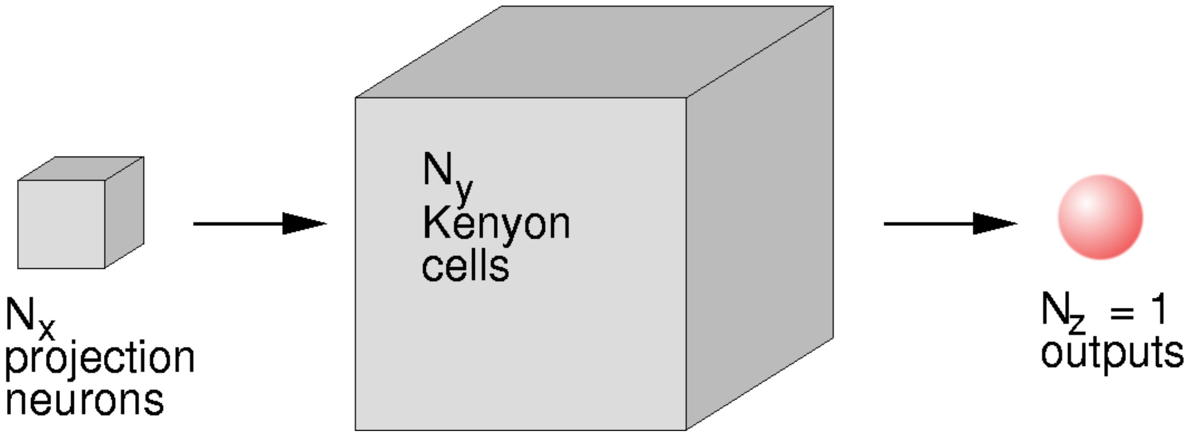


Classify one pattern from the rest

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



Example calculation

Probability for a Kenyon cell to be active, given n_x=k projection neurons fire

$$P(y_i = 1 \mid n_x = k) = \sum_{l=\theta}^{k} \binom{k}{l} p_{y\leftarrow x}^{l} \,(1 - p_{y\leftarrow x})^{k-l}$$

Probability for the number of active Kenyon cells, given …

$$P(n_y = r \mid n_x = k) = \binom{N_y}{r} P(y_i = 1 \mid n_x = k)^r \big(1 - P(y_i = 1 \mid n_x = k)\big)^{N_y - r}$$

Then the unconditioned probability is

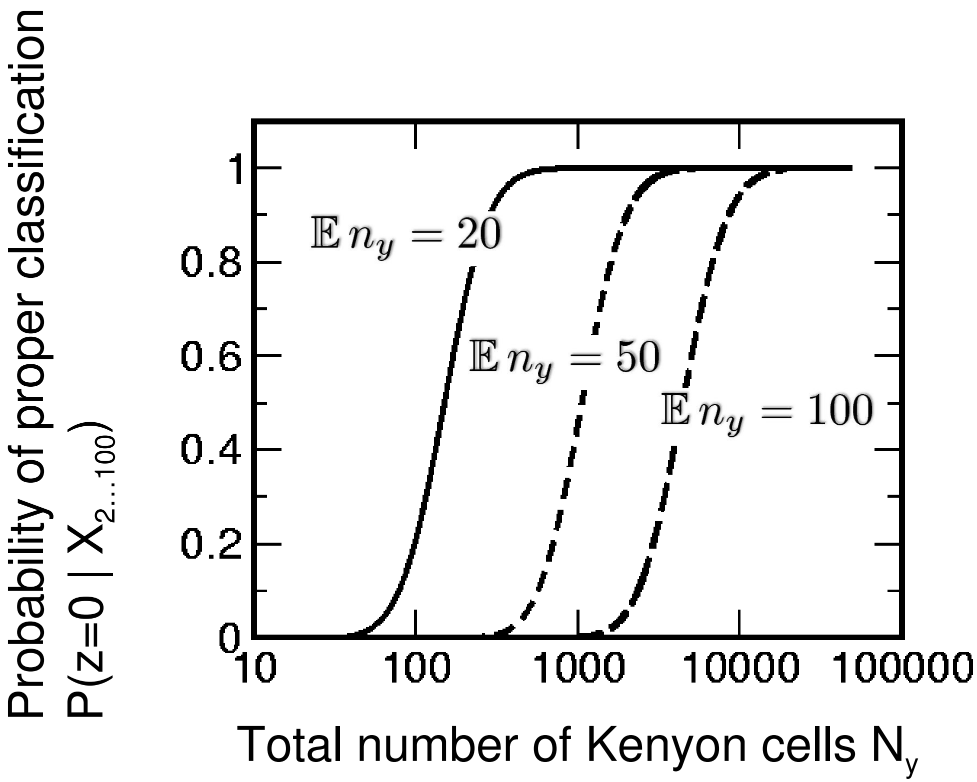
$$P(n_y = r) = \sum_k P(n_y = r \mid n_x = k)\; P(n_x = k)$$

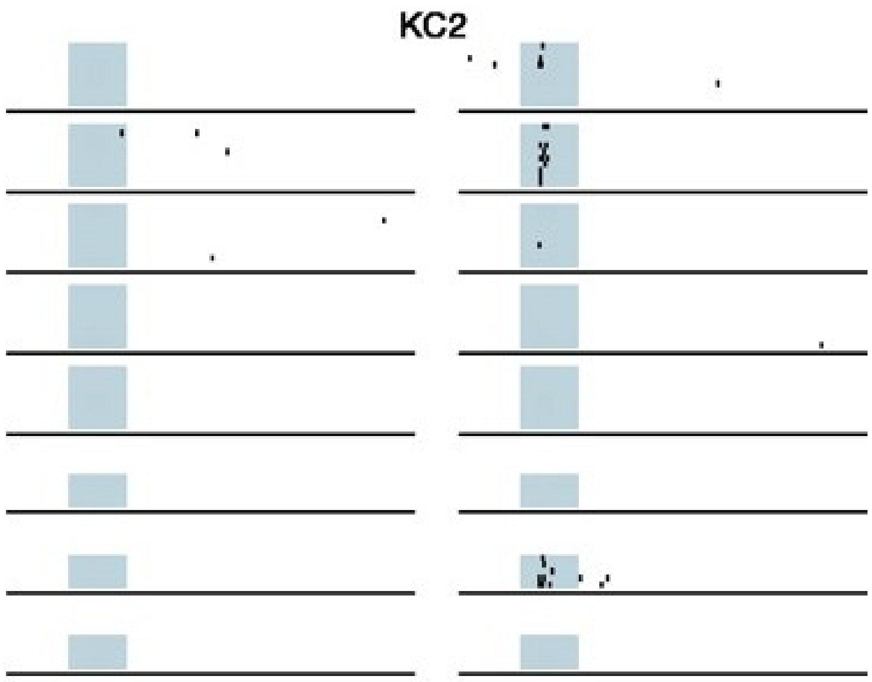
Leading (after some simplification) to

$$\mathbb{E}\, n_y = N_y\, p_x p_{y\leftarrow x}(1 - p_x p_{y\leftarrow x})$$

Example result: Classification needs sparse code

… and nature uses it!





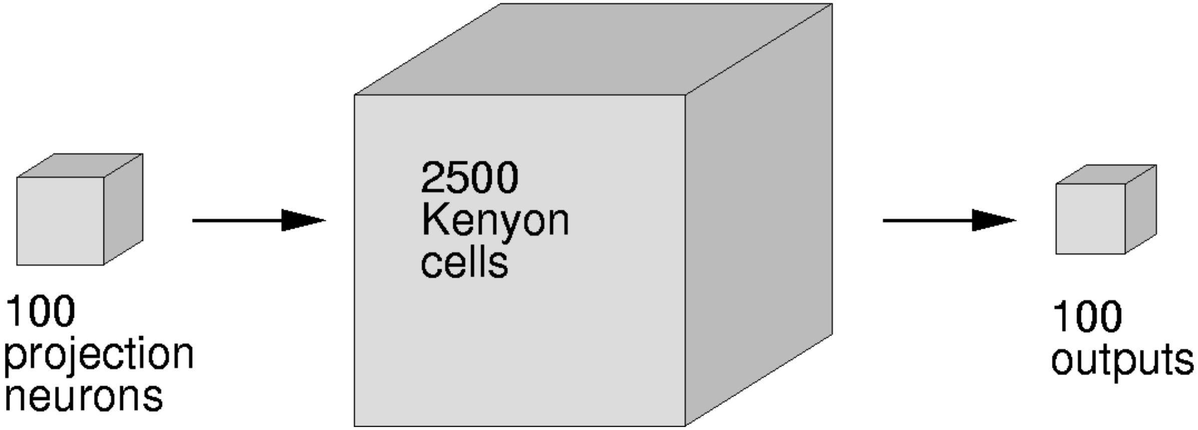
Perez-Orive et al., Science (2002)

"Have many, but only use a few"

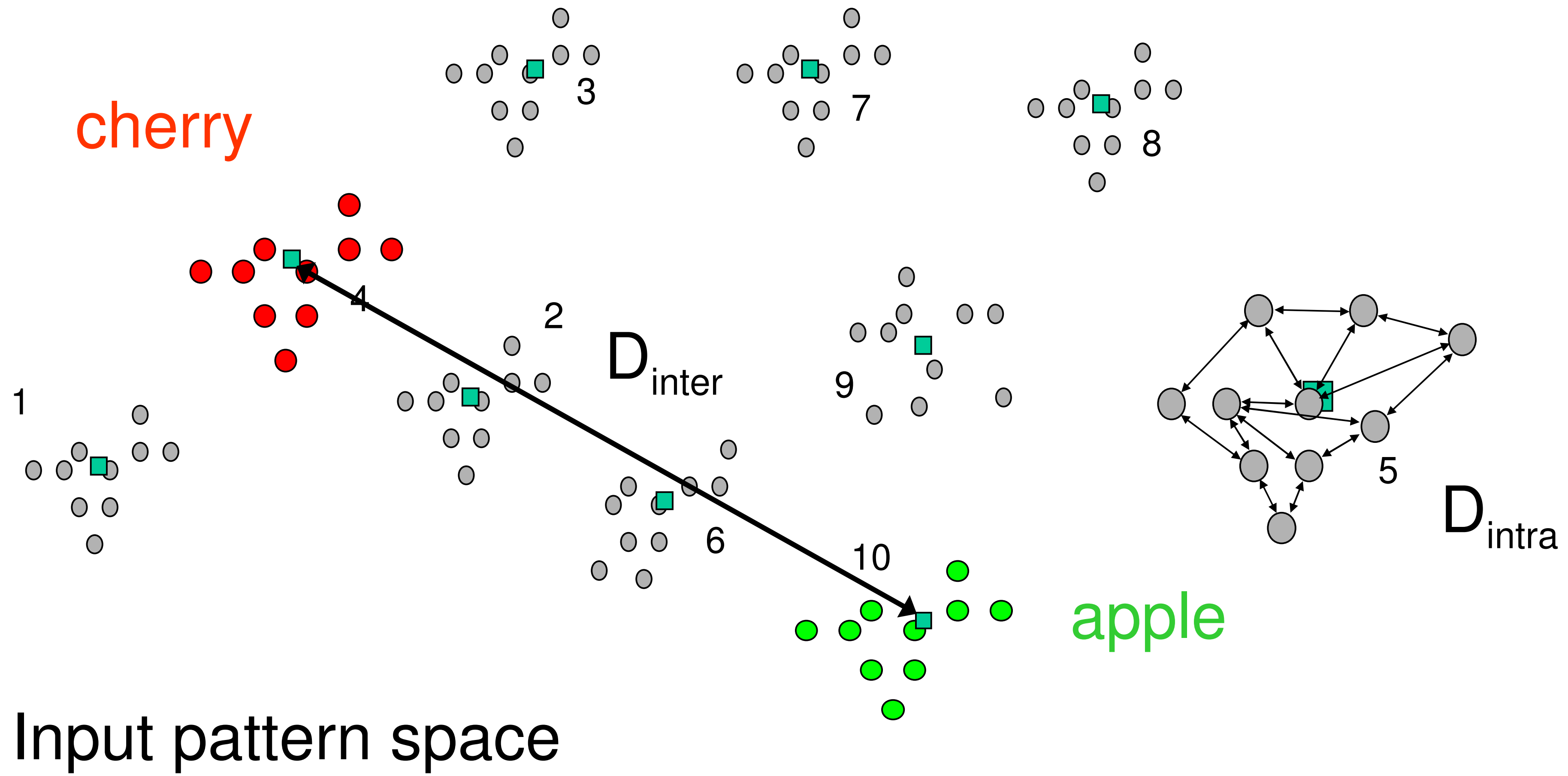
Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Classify classes of inputs

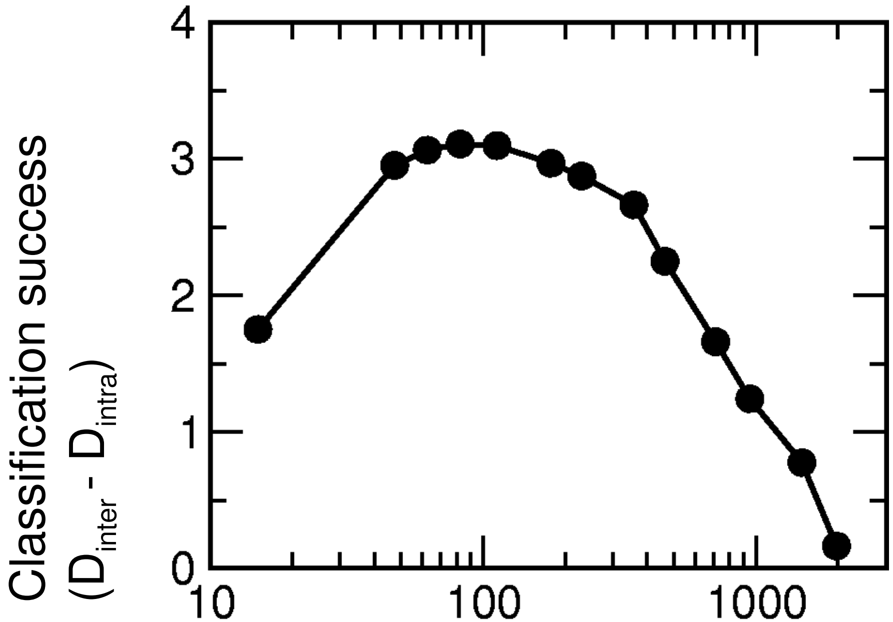
- 10 classes of inputs, 10 patterns each class
- "Winner-take-all" ouputs:
  The output neuron with the strongest input spikes
- Simulations in "*Drosophila* size"



Classes of input patterns



There are “optimal design parameters”



Number of active Kenyon cells $\mathbb{E}\, n_y$

$\exists$ Optimal $\mathbb{E}\, n_y$

of active Kenyon cells

Summary

- 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

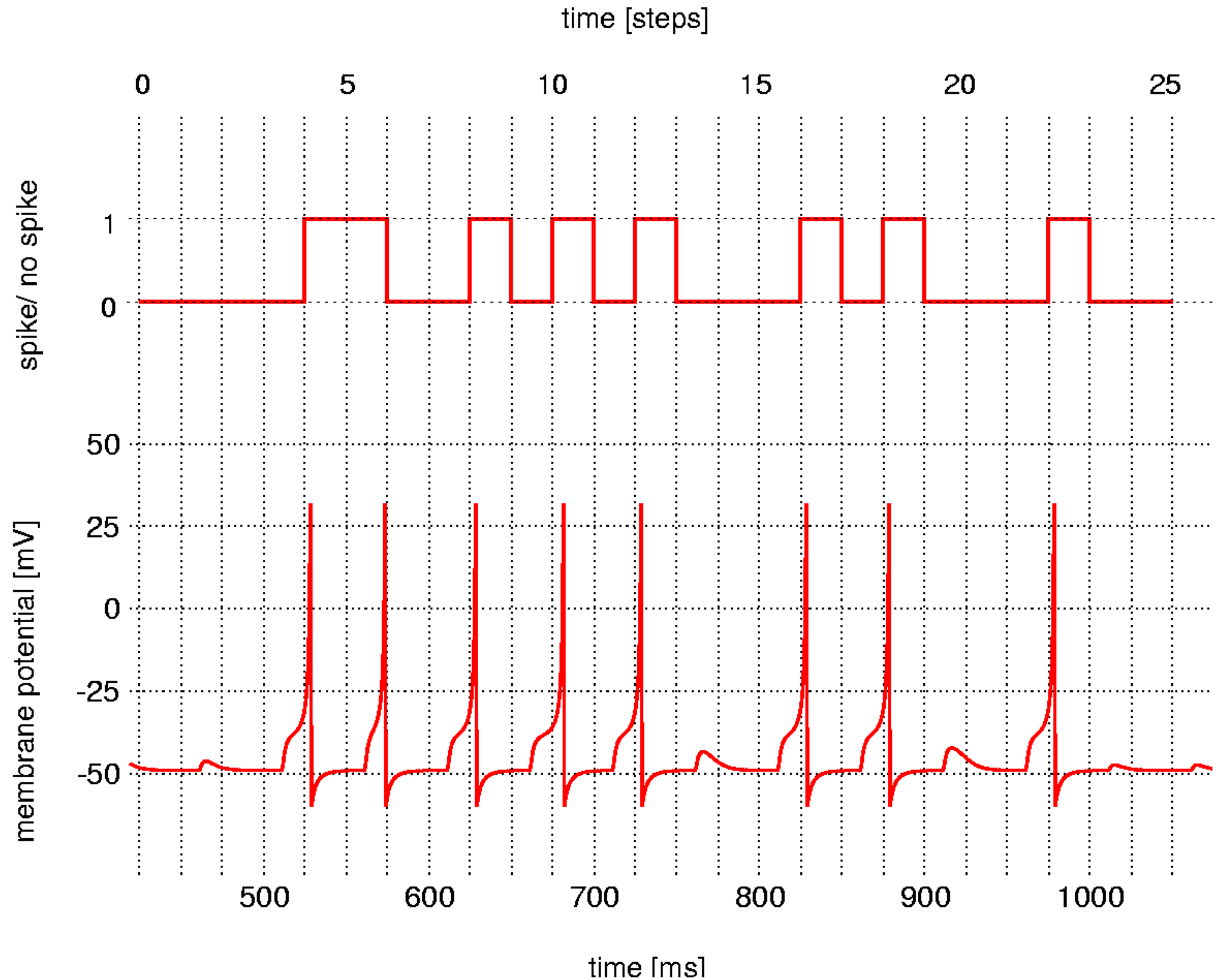
- The winner-take-all competition between output neurons has to be implemented artificially
- Gain control in the MB has to be implemented artificially

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

Spiking neuron models

McCulloch-Pitts

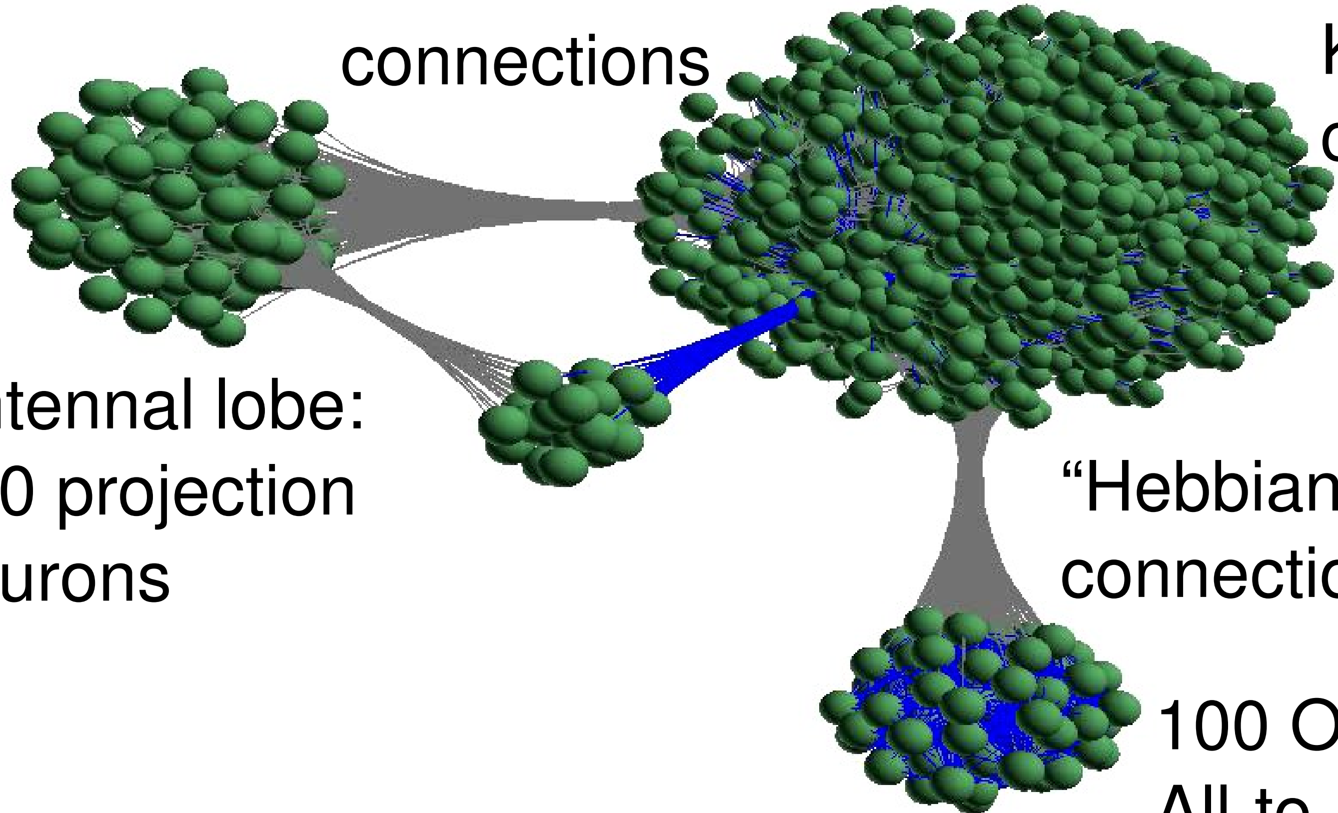
Spiking neurons



Process of recognition: Naïve locust

Random
connections

2500
Kenyon
cells



Antennal lobe:
100 projection
neurons

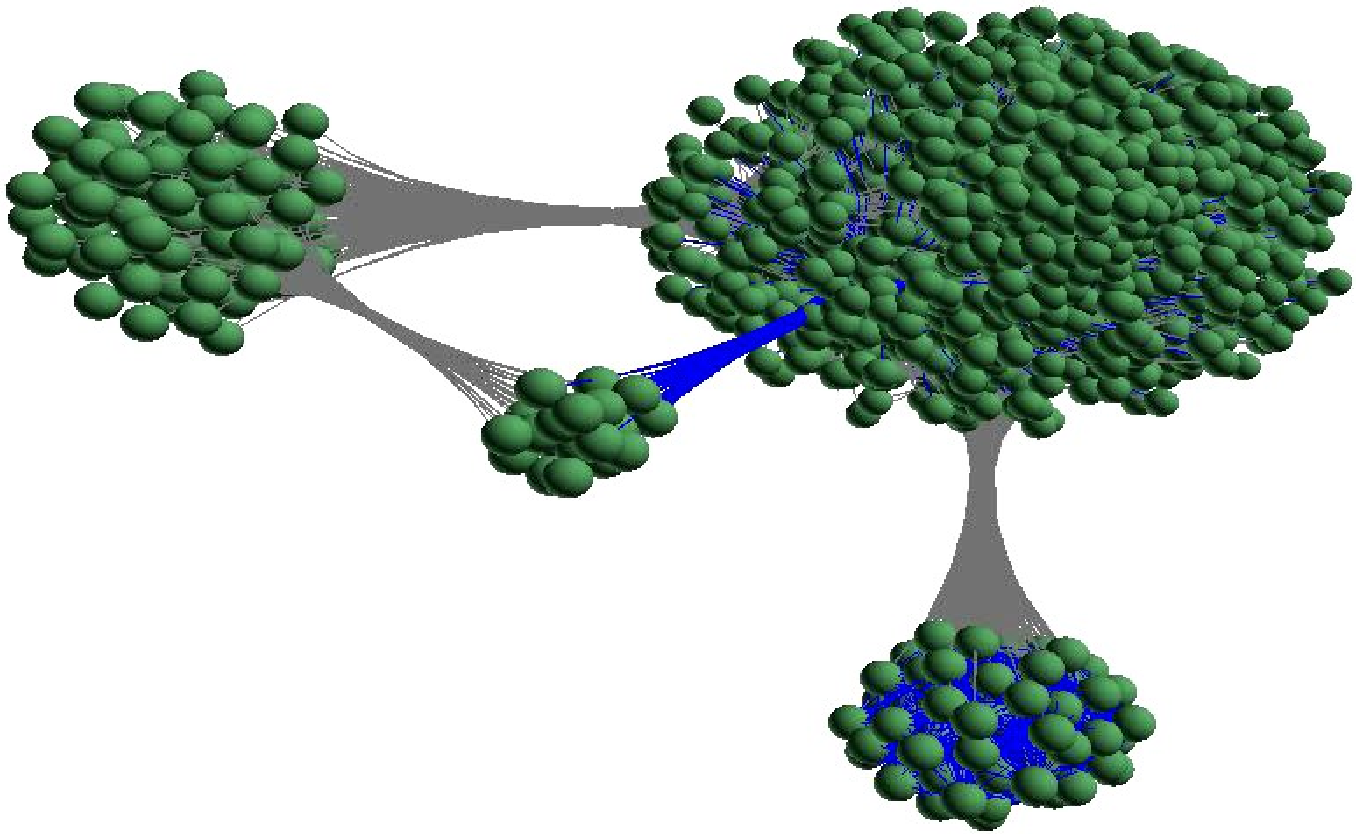
“Hebbian”
connections

100 Output neurons:
All-to-all inhibition

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

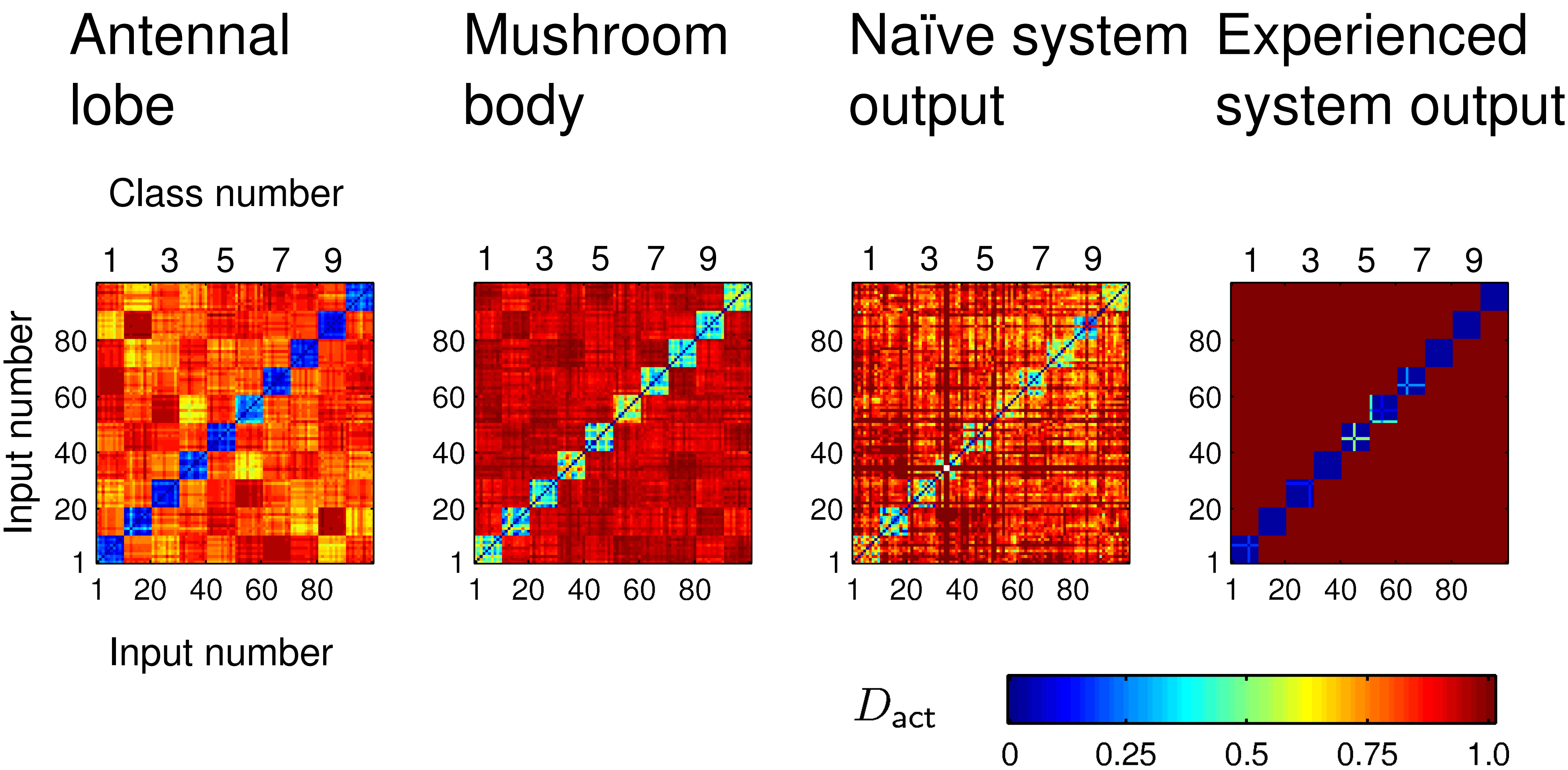
Dr. Thomas Nowotny,
Centre for Computational Neuroscience and Robotics
University of Sussex

Experienced locust



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

Quantitative Analysis



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

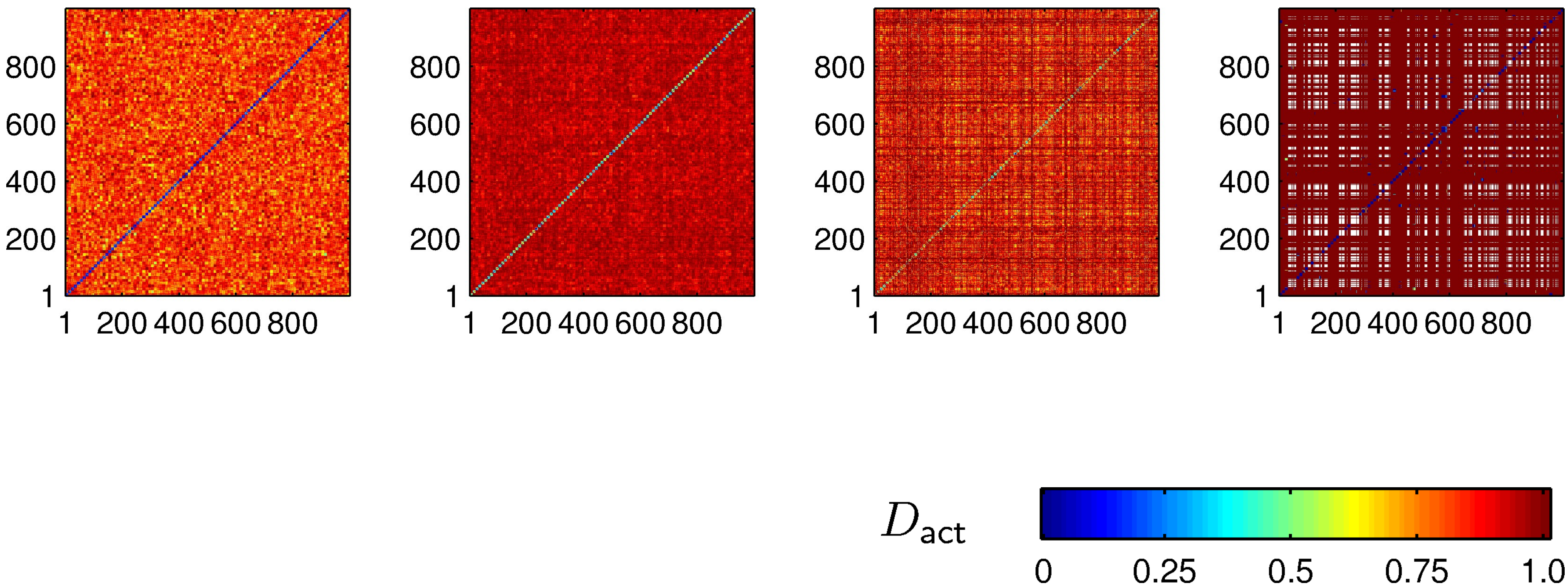
Quantitative Analysis

Antennal lobe

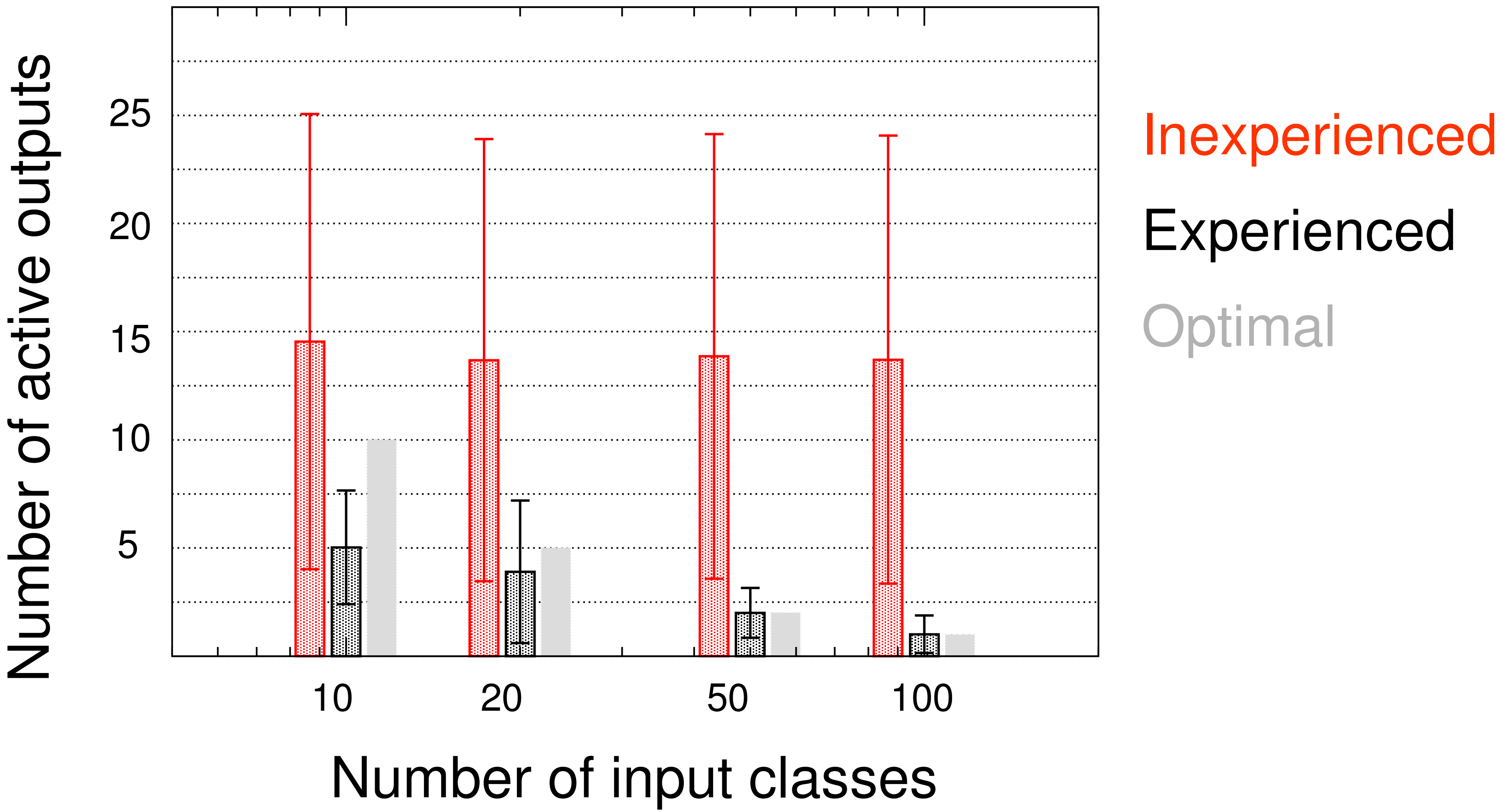
Mushroom body

Naïve system output

Experienced system output



Automatic detection of input set structure



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

Future directions

- Apply the McCulloch-Pitts/ spiking neuron "random SVM" to pattern classification problems (OCR, …)
  Huerta R & Nowotny T, Fast and robust learning by reinforcement signals: explorations in the insect brain, Neural Comput., in press
- Use of non-linear dynamics for *information preprocessing* in the antennal lobe
- Use of lateral excitation in the mushroom body to generalize to classification of *sequences*
- Feed-forward inhibition/ gain control
  Nowotny et al., in preparation

Future directions

- Reward systems and associative learning
- Other sensory input to mushroom bodies/ multimodal associations
- Implementations on massively parallel hardware
  Nowotny T, Muezzinoglu KM & Huerta R, Biomimetic classification on parallel hardware: Implementations in Nvidia¨ CUDA™, submitted