

MNS Project 2: Learning of Grid Cells

Claus Lang, C. Eren Sezener and Claudia Winklmayr

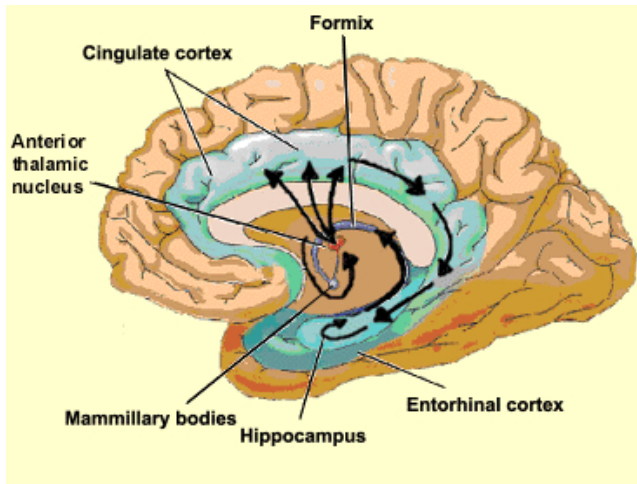
BCCN

February 9th 2016

Structure

- Introduction
- Modelling details
- Results

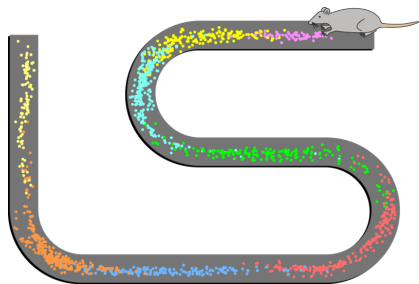
Introduction



In Hippocampus and the medial enthorhinal cortex (mEC) various types of neurons have been found that encode an animals spacial location.

Place cells

Located in hippocampus. Activated when the animal enters a specific region of the environment - the *place field*.

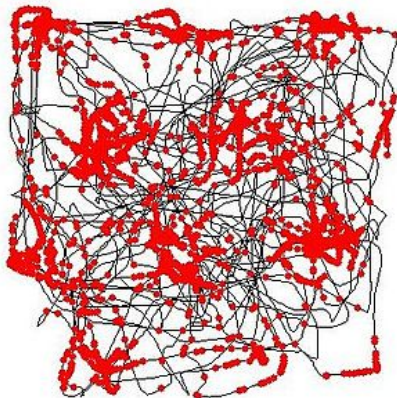


Reference:

J. O'Keefe and J. Dostrovsky, *The hippocampus as a spatial map. Preliminary evidence from unit activity in the freely-moving rat*. Brain Res. 34:171-175 (1971).

Grid cells

- Located in medial enthorhinal cortex (mEC). Activated at multiple spacial positions.
- Firing map shows hexagonal pattern.
- Mostly independent of visual stimulus
- Spacing, orientation, size of patterns.

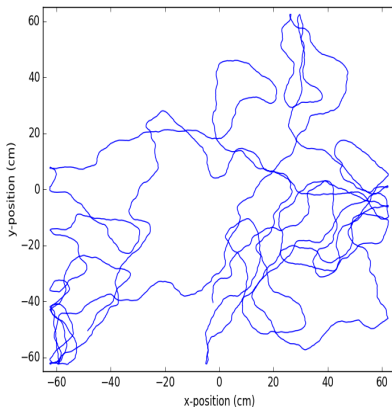


Reference:

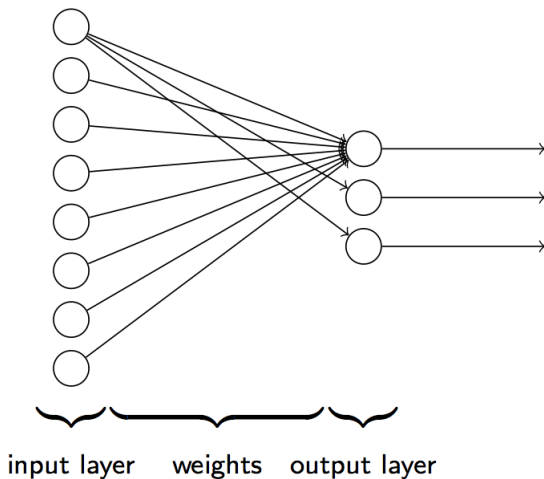
T. Hafting, M. Fyhn, S. Molden, M.-B. Moser and E.I. Moser, *Microstructure of a spatial map in the entorhinal cortex*. Nature 436: 801-806 (2005).

Rat trajectory

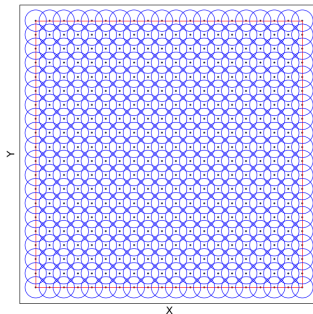
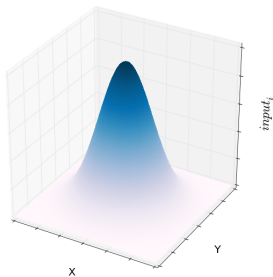
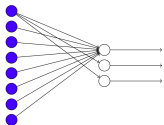
- Square environment of size 125×125 cm.
- Speed: $v = 0.4$ m/s
- Every 10ms: chose new direction from a gaussian distribution with
 - μ = previous direction
 - $\sigma = 0.2$



Model: Overview

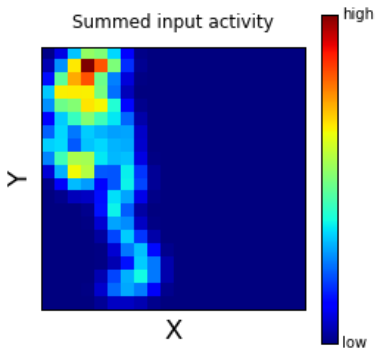
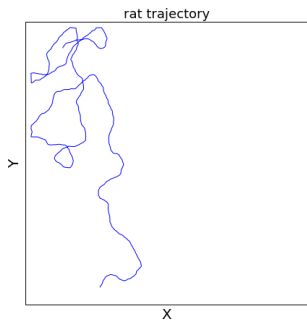
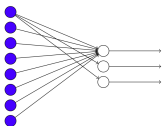


Model: Input Layer

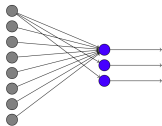


$$input_i = \exp\left(-\frac{\|rat_pos - center_i\|^2}{50}\right)$$

Model: Input Layer

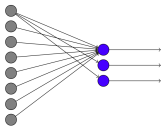


Model: Output Layer



$$h_j(t) = \sum_i w_{ij} \cdot input_i(t)$$

Model: Output Layer

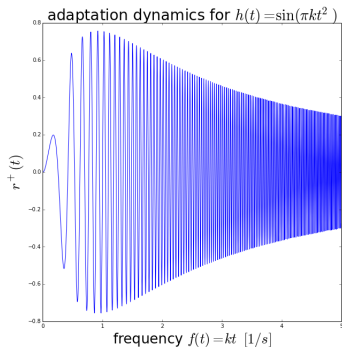


$$h_j(t) = \sum_i w_{ij} \cdot input_i(t)$$

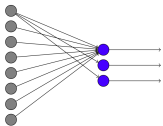
→ adaptation dynamics:

$$\tau^+ \frac{d}{dt} r_j^+(t) = h_j(t) - r_j^+(t) - r_j^-(t)$$

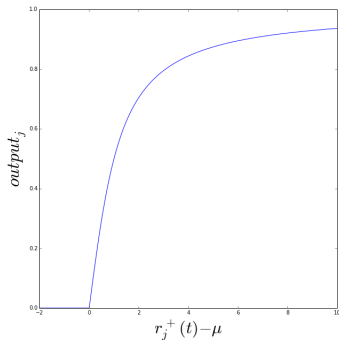
$$\tau^- \frac{d}{dt} r_j^-(t) = r_j^-(t)$$



Model: Output Layer



→ solution $r_j^+(t)$

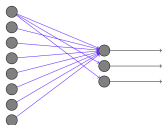


$$output_j(t) = F(r_j^+(t); g(t), \mu(t))$$

$g(t)$ - gain

$\mu(t)$ - threshold

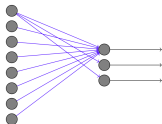
Model: Weight Updates



$$h_j(t) = \sum_i w_{ij} \cdot input_i(t)$$

How to determine the weights w_{ij} ?

Model: Weight Updates



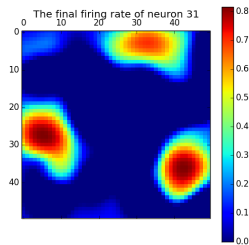
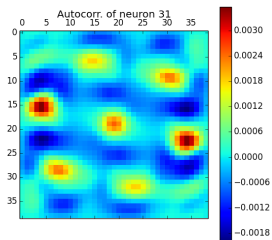
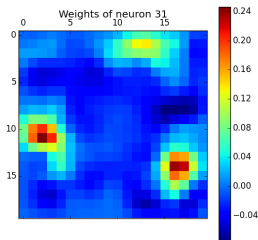
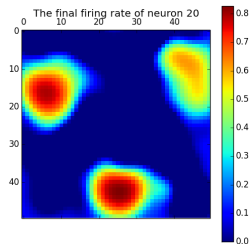
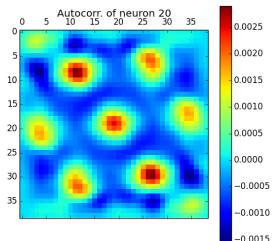
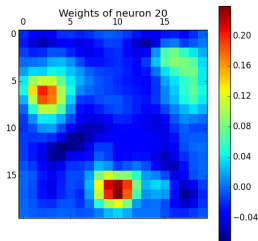
$$h_j(t) = \sum_i w_{ij} \cdot input_i(t)$$

How to determine the weights w_{ij} ?

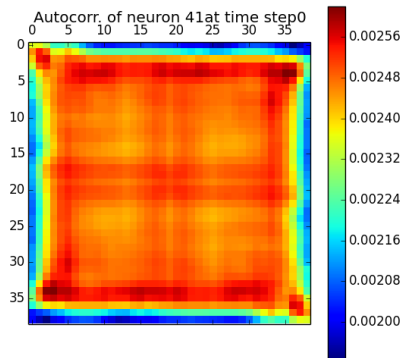
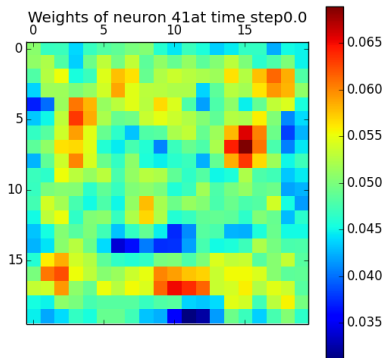
→ Hebbian learning dynamics ('Fire together, wire together')

$$w_{ij}(t + \Delta t) = w_{ij}(t) + \epsilon(input_i(t) \cdot output_j(t) - \overline{input_j} \cdot \overline{output_j})$$

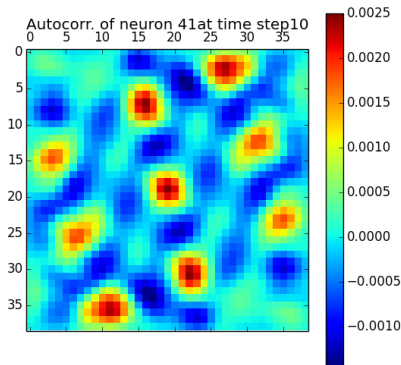
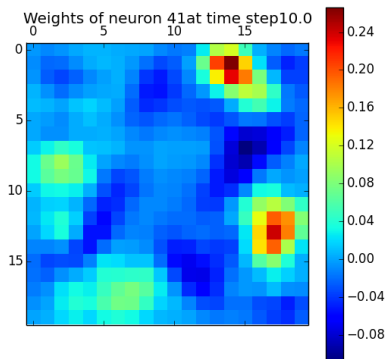
Cherrypicked final weights



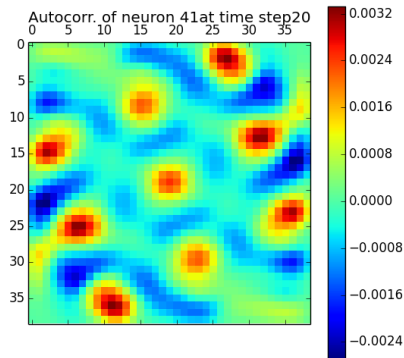
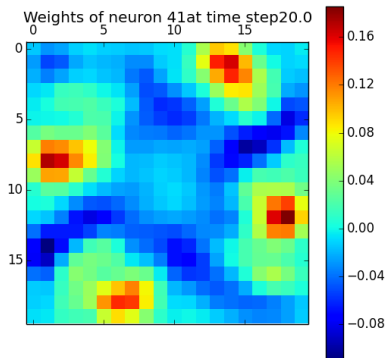
Time evolution of weights for a neuron



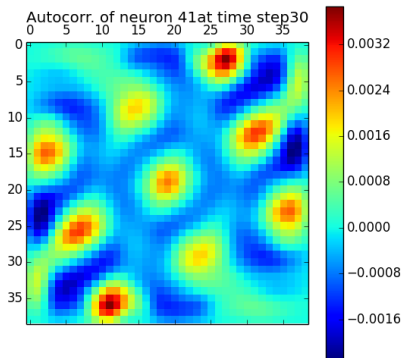
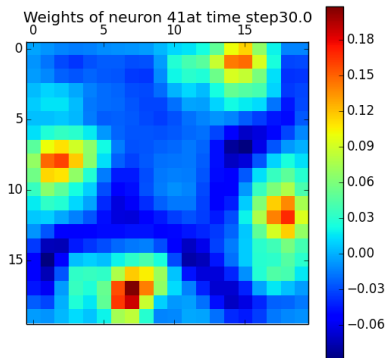
Time evolution of weights for a neuron



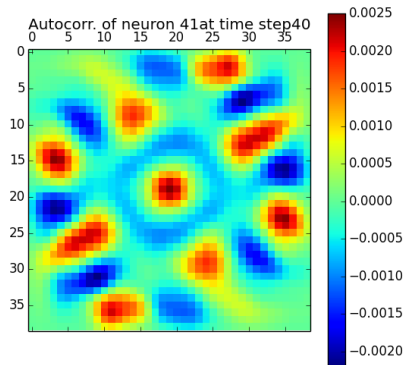
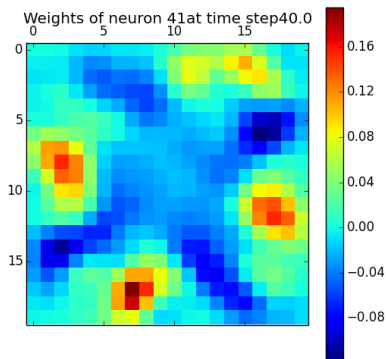
Time evolution of weights for a neuron



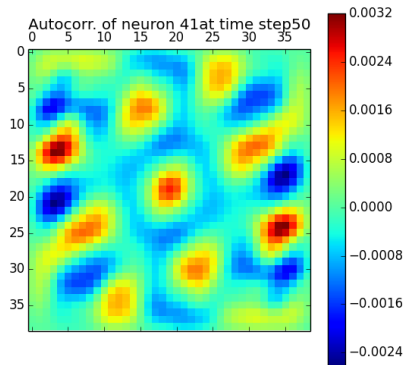
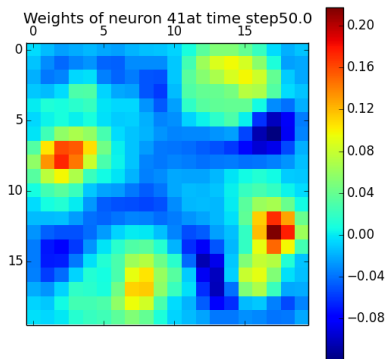
Time evolution of weights for a neuron



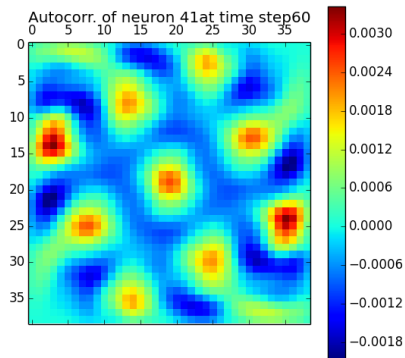
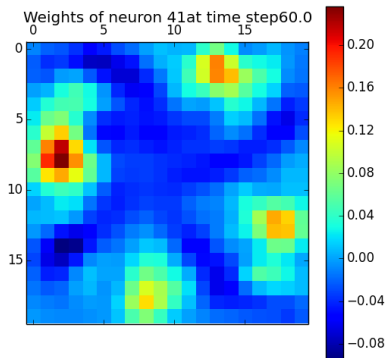
Time evolution of weights for a neuron



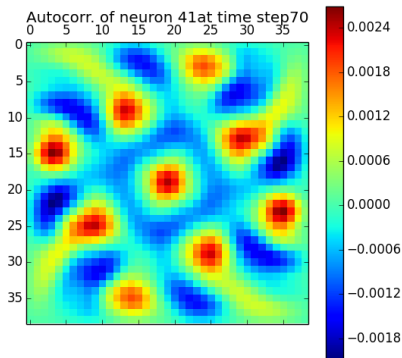
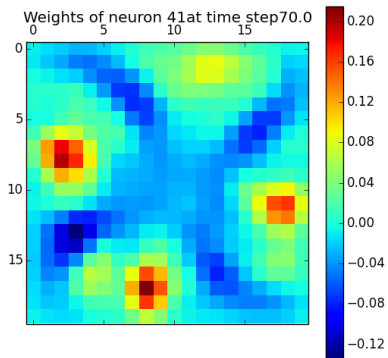
Time evolution of weights for a neuron



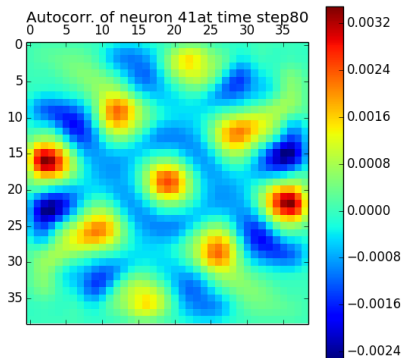
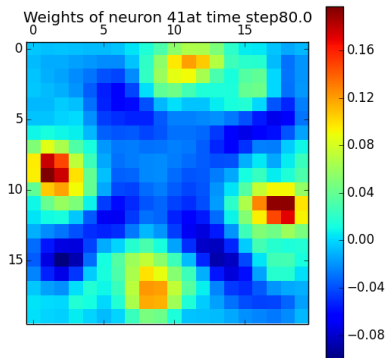
Time evolution of weights for a neuron



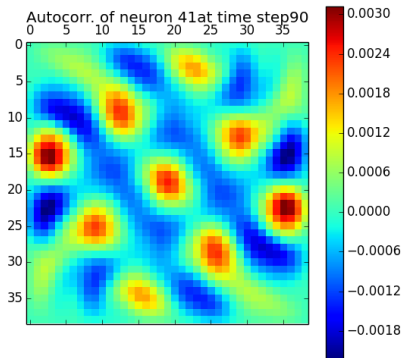
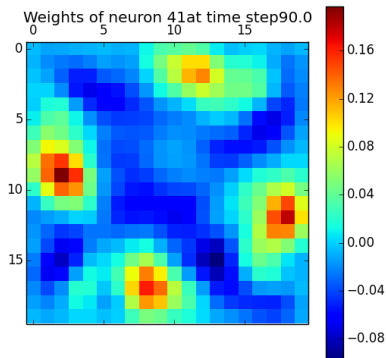
Time evolution of weights for a neuron



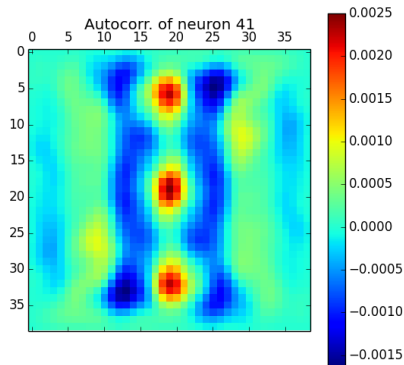
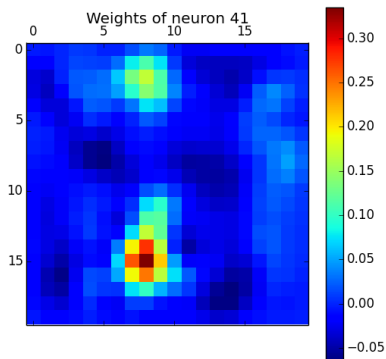
Time evolution of weights for a neuron



Time evolution of weights for a neuron



Time evolution of weights for a neuron



Thanks. Questions?