

Project 2: Learning of Grid Cells

Claus Lang, Can Eren Sezener, Claudia Winklmayr

09.02.2016

Abstract: Grid cells, located in rats medial entorhinal cortex (mEC), show increased firing activity when the animal enters specific regions of the environment and their firing maps show a characteristic hexagonal grid pattern. In our project we implement a computational model for the development of grid cells that was first introduced by Kropff and Treves in 2008 ([1],[2]). We simulate a rat exploring a square environment at constant speed and model the development of grid cell activity in a population of neurons receiving spacially tuned feed forward input.

1 Introduction

Different types of neurons encode the spatial location of an animal. Place cells - located in hippocampus - were discovered in the 1970s by O'Keefe and Dostrovsky [3], they show increased activity when the animal enters a specific region of the environment - the so-called place field.

Grid cells - located in medial entorhinal cortex (mEC) were then discovered in 2005 [4]. In contrast to place cells, grid cells are activated at multiple locations and their firing maps show a characteristic hexagonal grid pattern. Grid cells can be characterized by spacing, orientation and spatial phase. Neighbouring grid cells show similar spacing and orientation.

It was proposed that grid cells perform a path integration task, taking into account the rat's current position, its running speed and direction of movement. Sensual information, on the other hand, was attributed a secondary role in the formation of the grid patterns [5].

2 Model

The Kropff and Treves model assumes place cells as the basis for grid cell development. In our implementation the model consists of a layer of 400 input neurons (simulating place cells) and 100 output neurons (simulation grid cells). Every unit of the input layer is connected to every unit of the output layer. This is schematically visualized in figure ???. At each time step the place cells receive spatially tuned input encoding the rat's current position. Every output unit then receives input from all place cells and the weights connecting input and output neurons are updated by a Hebbian learning rule.

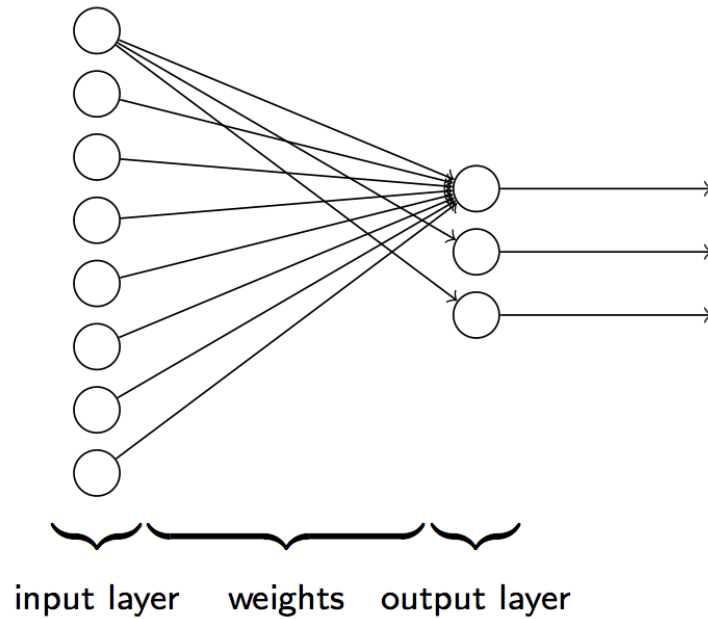
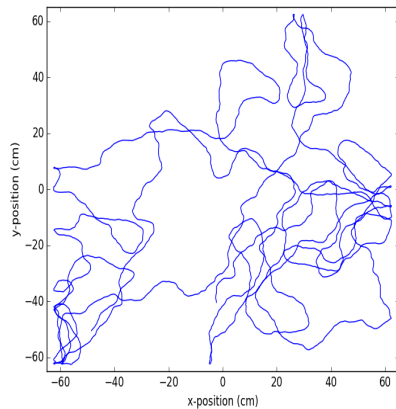


Figure 1: The feed-forward network (showing only a subset of the nodes and only all connections from one input node and to one output node)

2.1 Spatial exploration



To create the spatial coordinates that serve as input for the first layer of neurons, we simulate a rat exploring a square environment of size $125 \text{ cm} \times 125 \text{ cm}$ at a speed of 0.4 m/s . At each time step $\tau = 10 \text{ ms}$ we chose a new direction by randomly drawing an angle ϕ from a normal distribution, which is centred around the direction at the previous time step and has a standard deviation of 0.2 rad .

Figure 2: trajectory of the virtual rat exploring for 50 seconds

2.2 Input layer

The input layer consists of 400 place units processing information about the rats location $\mathbf{x}(t)$ at a given time t (see figure ??). The firing $r_j^{in}(t)$ of each place unit is modelled by a Gaussian function centred around the cell's preferred location \mathbf{x}_j . The width of the

firing field is denoted by $\sigma_p = 5$ cm.

$$r_j^{in}(t) = \exp\left(\frac{-|\mathbf{x}^t - \mathbf{x}_j|}{2\sigma_p^2}\right) \quad (1)$$

In figure 2 we can see the activation of input layer neurons as the rat explores a small section of the environment:

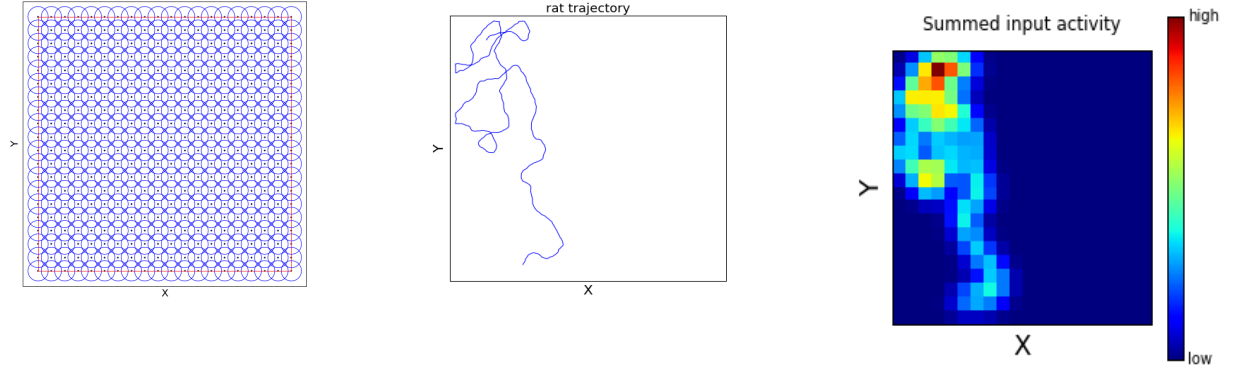


Figure 3: From left to right: (i) Distribution of place fields across the environment, (ii) Trajectory of the virtual rat and (iii) Activity of place cells summed over exploring time. Each pixel corresponds to one place cell (arranged as in (i)).

2.3 Output layer

At each time step output unit i receives as input a weighted sum h_i of the place cell activities:

$$h_i(t) = \sum_j w_{ij} r_j^{in}(t). \quad (2)$$

By w_{ij} we denote the weight of the connection from the j -th input unit to the i -th output unit. $h_i(t)$ is then subject to a dynamic adaptation process involving the activation variable r^+ and the inactivation variable r^- . The speed of the adaptation is governed by the time constants $\tau^+ = 0.1$ s and $\tau^- = 0.3$ s:

$$\tau^+ \frac{d}{dt} r_i^+ = h_i(t) - r_i^+(t) - r_i^-(t) \quad (3)$$

$$\tau^- \frac{d}{dt} r_i^- = h_i(t) - r_i^-(t) \quad (4)$$

In order to understand the dynamics of this system of differential equations, consider a sinusoidal input of $h_i(t) = \sin(\pi k t^2)$. Figure ?? shows the numerical solution of r_i^+ in that case plotted versus the frequency of the input kt . The plot shows that there is a resonance frequency of about $0.92 \frac{1}{s}$, where the activation r_i^+ is maximal.

The variable r^+ is then used to calculate the final output via the following activation function:

$$r_i^{out}(t) = \frac{2}{\pi} \arctan(g(r_i^+(t) - \mu)) \Theta(r_i^+(t) - \mu). \quad (5)$$

The Heaviside step-function Θ ensures that all output values are nonnegative and the factor $\frac{2}{\pi}$ serves to normalize them to values between 0 and 1.

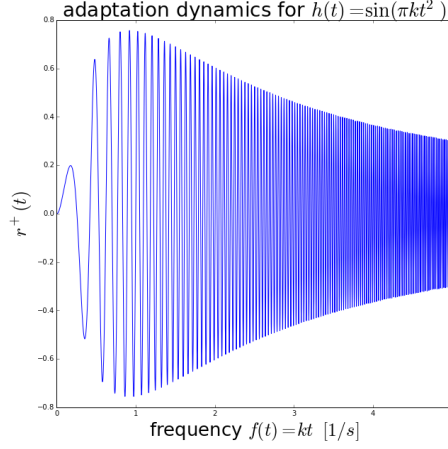


Figure 4: numerical solution of adaptation dynamics for sinusoidal input

The two control parameters g and μ are the gain and the threshold of the system. g serves to keep the mean activity across neurons - defined via $a = \frac{1}{N} \sum_i r_i^{out}$ - within a 10%-range from the value a_0 . The parameter μ respectively controls the sparsity $s = \frac{(\sum_i r_i^{out})^2}{\frac{1}{N} \sum_i (r_i^{out})^2}$ and keeps it within a 10% error bound of the value s_0 . In both above expressions the parameter N denotes the number of output neurons. We determine g and μ by an optimization algorithm where at each step they are updated in the following way:

$$\mu^{t,l+1} = \mu^{t,l} + b_\mu(a^l - a_0) \quad (6)$$

$$g^{t,l+1} = g^{t,l} + b_g g^{t,l}(s^l - s_0) \quad (7)$$

l indicates the step of the iteration. The updating process is stopped if both a and s fall within the 10 % error bound of a_0 and s_0 or when an upper bound of 100 iterations is reached.

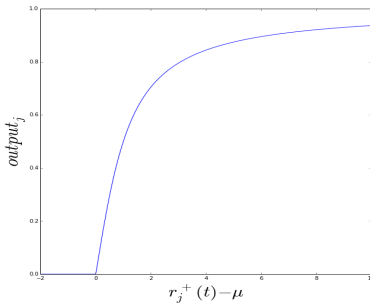


Figure 5: Activation function for output neurons

a_0	s_0	$\mu^{0,0}$	$g^{0,0}$	b_μ	b_g
0.1	0.3	0	4.5	0.01	0.1

Table 1: Default parameters for adaptation dynamics

2.4 Weights

The weights between the input and the output layer are initially being drawn from a uniform distribution in $[0, 1]$ and normalized to unit norm i.e. for every output unit i we

impose $\sum_j w_{ij}^2 = 1$. At each timestep of the simulation the weights are then updated using a Hebbian learning rule:

$$w_{ij}(t + \Delta t) = w_{ij}(t) + \epsilon(r_i^{out}(t)r_j^{in}(t) - \bar{r}_i^{out}(t)\bar{r}_j^{in}(t)), \quad (8)$$

$\epsilon = 0.005$ is a moderate learning rate and $\bar{r}_i^{out}(t)$ and $\bar{r}_j^{in}(t)$ are the running averages of input and output firing rates.

$$\bar{r}_i^{out}(t) = \bar{r}_i^{out}(t-1) + \eta(r_i^{out}(t) - \bar{r}_i^{out}(t-1)) \quad (9)$$

$$\bar{r}_j^{in}(t) = \bar{r}_j^{in}(t-1) + \eta(r_j^{in}(t) - \bar{r}_j^{in}(t-1)). \quad (10)$$

$\eta = 0.05$ sets the time averaging constant. After each update the weights are kept normalized to unit norm.

3 Results

We ran the simulation for 2.5 hours in rat time. The results show that this duration is sufficient for grid cells to develop. Figure 8 shows the time evolution of weights of a grid cell. The cell we picked has one of the most apparent grid structure among the other cells.

When all grid cells are inspected, it is seen that most grid cells have similar spatial frequencies, however, their orientations and off-sets are different. Figure 9 illustrates this difference.

Project 2: Learning of Grid Cells

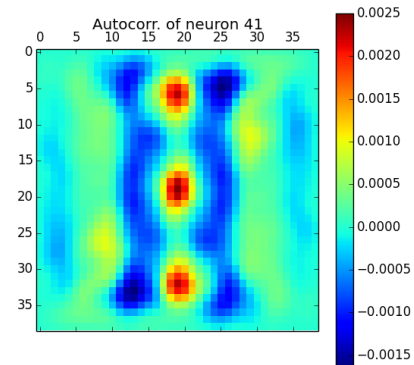
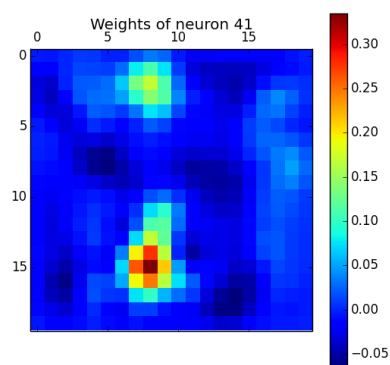
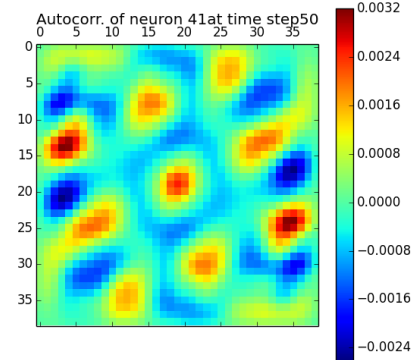
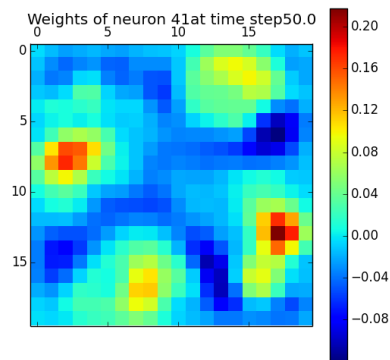
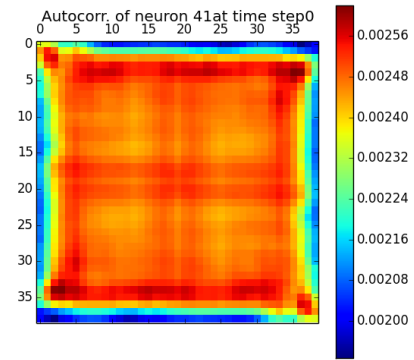
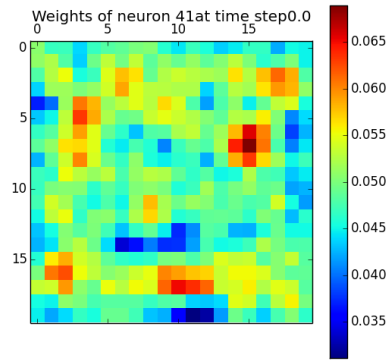


Figure 6: weight development

Figure 7: autocorrelation development

Figure 8: The weights and the autocorrelogram of a neuron at the beginning, middle and the end of the simulation

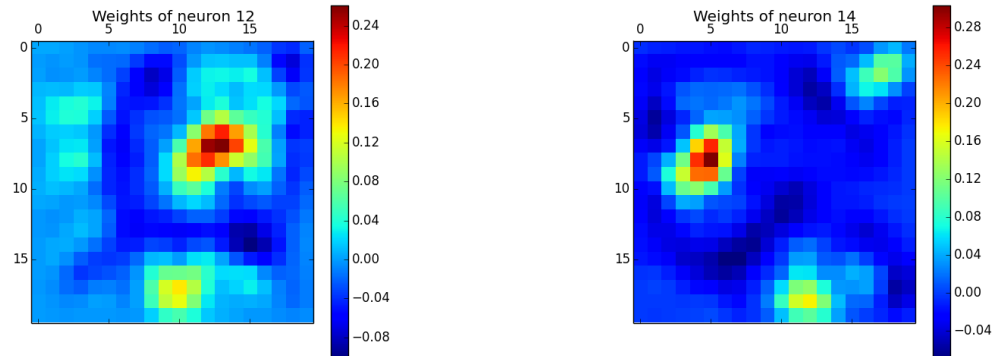


Figure 9: Different grid cells have different orientations and off-sets but similar spatial frequencies.

4 References

1. E.Kropff and A.Treves, *The Emergence of Grid Cells: Intelligent Design or Just Adaptation?*, Hippocampus 18: 1256-1269 (2008).
2. B. Si, E. Kropff, and A. Treves, *Grid alignment in entorhinal cortex*. Biological cybernetics 106: 8-9 (2012).
3. 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).
4. 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).
5. J.S.Barlow, *Inertial navigation as a basis for animal navigation*. J Theor Biol. 6: 76-117 (1964)