Cognitive Maps for Navigation in Open Environments

Verena Vanessa Hafner
AI Lab, Dept. of Information Technology, University of Zurich,
Winterthurerstr. 190, 8057 Zurich, Switzerland

Abstract. This paper presents a computational model which explains the formation of cognitive maps similar to those found in mammals and may be used for navigation tasks in open environments (as opposed to mazes). The model is inspired by the discovery of place as well as head direction cells in the rat's hippocampus. The modelled cognitive maps consist of mainly topological relations between 'place cells' and are generated by a modified version of the Kohonen self-organizing map. A model which simulates physical forces as in natural systems is applied to extend the relatively sparse topological map with metric information. Both the cognitive map and the force model are closely related to functions that could be performed in the brain. The performance of the model is tested in navigation experiments with a simulated robot in an open environment. The robot is equipped with distance sensors and a compass, but has no information about its current coordinate position.

1 Introduction

This work is inspired by the discovery of place cells as well as head direction cells in the rat's hippocampus. While insects seem to use only a guidance and orientation system for navigation, mammals are thought to have a mapping system which requires the use of spatial memory. This neural mapping system is called a *cognitive map*, a term first used by Tolman in 1948 [10]. In the rat's hippocampus, cells were detected which preferably fired when the rat was in a particular portion of its environment, but were largely independent of its orientation and actual view (O'Keefe, 1976 [6]). The area in the environment where one particular place cell fires at the highest rate is called the *place field* of this cell. Another type of cells were found which preferably fired when the rat's head was turned in a certain direction, regardless of its position in the environment (Taube et al., 1990 [9]). By exploring the environment for a certain amount of time, a rat creates a representation of this environment, in other words, a map in its brain. After the map has been created, the sensor input can even be removed (e.g. the light switched off), yet the proper place and head direction cells still fire during movement (O'Keefe and Speakman, 1987 [7]). Eventually, the place as well as the head direction representation shifts slightly (Knierim et al., 1995 [3]) but in a coordinated manner, which shows that place and head direction cells are somehow correlated. These findings suggest that a mental representation of the environment is stored in memory which also contains information about the direction of paths between a pair of places.

A neural network model of a navigation system is described which takes into account these findings. It will be presented in section 2. The neurons and synapses in this model are idealised and should not be regarded as biologically realistic neurons and synapses. However, they do share some properties with their biological correlates. In section 3, a physical force model will be presented which resolves the conflict between topological and metric map representations. This model allows for both a flexible topological map and the advantage of the higher information content of metric maps.

The navigation scenario presented in section 4 consists of a simulated environment which a robot explores by moving through it and receiving sensor feedback. The information of both the sensor inputs and the movement direction is fed into the neural network, which represents the cognitive map. The results will be discussed in section 5.

2 Neural Network Model

In this section, the computational neural network model of the cognitive map is described. Its function is to create a topological map of an environment explored using sensor data as an input to the neural network. The self-organizing cognitive map consists of a modified Kohonen Network [4]. It has some similarities with the self-organizing sequence map implemented by Schölkopf and Mallot [8], who explored view-based navigation in mazes.

The main difference between this model and the standard Kohonen self-organizing map is that no assumption about the spatial position of the neurons in the cognitive map is made, and therefore only the connection weights of the winning neuron are updated. As there has not yet been found any obvious correlation between the spatial arrangement of place cells in the hippocampus and the spatial arrangement of their place fields, this change seems more appropriate. Another important change to the standard Kohonen network is the enforcement of the weights between the current winner neuron and the winner neuron from one time-step before. This action is based on the continuous movement of the agent in its environment. It allows for the creation of a topological map without any association between place cells and spatial coordinates.

The neural network consists of an input layer $\mathbf{f} = (f_1, ..., f_I)$ with I neurons and an output layer $\mathbf{o} = (o_1, ..., o_J)$ with J neurons, which are fully interconnected by weight vectors $\mathbf{r_j}, (j \in \{1, ..., J\})$. The output layer (or map layer) is also fully interconnected with itself by connection weights α_{jk} and angles $\rho_{jk}, (j, k \in \{1, ..., J\})$. A preprocessed input vector of a place is fed into the input layer and triggers a "winner takes all" process in the output layer. The activation a_k^t for each neuron o_k in the map layer is calculated using the sum x_k^t of four different terms within a sigmoid function $a_k^t = g(x_k^t) = (1 + e^{-x_k^t})^{-1}$ with

$$x_k^t = s_k^t + c_k^t + l_k^t - \theta \qquad (1)$$

The terms, which will be described in more detail in the following subsections, are the feature similarity s_k, the connectedness c_k, the movement values l_k and a fixed threshold θ. The threshold θ is subtracted to shift the input of the sigmoid function more towards negative values, which results in a greater variety of possible activation values. The output neuron o_j with the strongest activation is deemed the winner neuron. Its weight vector $\mathbf{r_j}$, as well as its connectedness and movement value to the last winner, will be updated.

2.1 Feature Similarity

The feature similarity s_k^t is a measure of resemblance between the received input and an already perceived view, without the context being taken into account. It is measured by the product of the input vector $\mathbf{f}$ and the weight vector $\mathbf{r_k}$ connecting $\mathbf{f}$ to a neuron o_j in the output layer, $s_k^t = \sum_{i=1}^{I} (f_i^t r_{ik}^t)$. Since $\mathbf{f}$ as well as $\mathbf{r_k}$ have been normalized, it reaches its

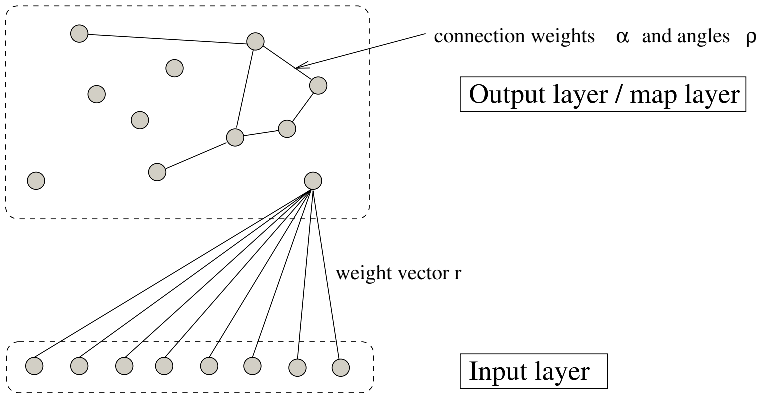

Figure 1: The neural network structure. The neurons in the input layer are fully connected to the map layer neurons. The map layer neurons are fully interconnected from the beginning of the learning process, but initially have zero weight.

maximum value when $\mathbf{f} = \mathbf{r_k}$. The weights $\mathbf{r_k}$ of the winner neuron are updated according to the Kohonen learning rule

$$\mathbf{r_k^t} = \mathbf{r_k^{t-1}} + \delta(\mathbf{f^t} - \mathbf{r_k^{t-1}})$$

where δ is a learning constant. The weight vectors are then normalized to constant length. This training law gradually aligns the weight vectors $\mathbf{r_k}$ of winning neurons o_k with the direction of the normalized input vectors $\mathbf{f}$.

2.2 Connectedness

The connectedness c_k^t is used to bring into account where the network 'thinks' it has just been, as well as the (un)certainty of a connection between two places. The connection value α_{jk} between the winner neuron o_k and the last winner neuron o_j increases according to the following formula:

$$\alpha_{jk}^t = \alpha_{jk}^{t-1} + \gamma(\alpha_{\text{max}} - \alpha_{jk}^{t-1}) \tag{2}$$

where γ is a learning constant and α_{max} is a constant maximum connection weight. In the activation function, the connectedness c_k^t of each neuron o_k in the output layer to neurons o_j is weighted with the last activation a_j^{t-1} of the neurons.

$$c_k^t = \sum_{j=1}^{J} a_j^{t-1} \alpha_{jk}^t$$

2.3 Movement Value

As additional information, an estimation of the direction in which the agent has just moved is generated. The length of the travelled path from the last place is unknown and can only be estimated as being the same for all connected places. In a biological organism, direction estimation could be taken by using global landmarks as a compass. This direction sense is represented by the head direction cells in rats.

The movement angle between two nodes is stored in ρ_{jk} and is updated each time step when o_k is the winner neuron after o_j by bisecting the stored angle ρ_{jk}^{t-1} and the input angle ρ_{inp}^t. The movement value m_{jk} for each node is calculated as

$$m_{jk} = \cos\left(|\rho_{jk} - \rho_{\text{inp}}|\right).$$

The value for $|\rho_{jk} - \rho_{\text{inp}}|$ lies between 0 and 2π which modifies m_{jk} to a value between -1 and 1.

In the activation function, the movement value of the connection between neuron o_k and each neuron o_j in the output layer is weighted with the last activation a_j^{t-1} of the neurons.

$$l_k^t = \sum_{j=1}^{J} a_j^{t-1} m_{jk}^t$$

In a more biologically plausible model, the angle information could be modelled separately, maybe in the form of head direction cells influencing the weights between place cells. Processes like these have been observed in the brain and are known as *presynaptic inhibition* or *facilitation*.

By inserting the input vector $\mathbf{f}$, the weight vectors $\mathbf{r_k}$, the last activations a_k^{t-1}, the map layer weight vectors $\alpha_{\mathbf{k}}$, and the movement weight vectors $\mathbf{m_k}$, x_k^t is expanded to

$$x_k^t = \sum_{i=1}^{I} f_i^t r_{ik}^t + \frac{1}{J} \sum_{j=1}^{J} \left(a_j^{t-1}(\alpha_{jk}^t + m_{jk}^t)\right) - \theta \tag{3}$$

3 Physical Force Model

The model introduced in this section allows the extraction of metric information from the map presented by the neural network in section 2. With this force model, a flexible and inexpensive topological representation can be preserved, but can also be extended with metric information when required. To get a better idea of the cognitive map, it is useful to arrange it as a graph $G = (V, E)$, where the set of vertices V represents the map layer nodes, and the set of edges E represents the connections between nodes. The only information the cognitive map contains apart from its topology is angle information.

Between all connected nodes, equally strong springs are assumed. The model makes use of the assumption that all edges have approximately the same length. The nodes are imagined as repulsive charges and the edges as springs between them. Let $\delta(v, w) \in \mathrm{R}^2$ be the distance vector between node v and node w and $d(v, w) = ||\delta(v, w)||$ its Euclidean distance. Between each pair of nodes, there is a force $F(v, w) = F_{\text{rep}}(v, w) + F_{\text{attr}}(v, w)$, the sum of a repulsive and an attractive force. The repulsive force is caused by the charges and the attractive force is caused by the springs. If the two nodes are not connected by a spring, then $F_{\text{attr}}(v, w) = 0$.

The attractive force is computed according to Hooke's law:

$$F_{\text{attr}}(v, w) = \gamma_{\text{attr}} \delta(v, w)$$

and the repulsive force is set to be proportional to the inverse square of the distance,

$$F_{\text{rep}}(v, w) = -\gamma_{\text{rep}} \frac{\delta(v, w)}{d(v, w)^3} = -\gamma_{\text{rep}} \frac{\delta(v, w)}{||\delta(v, w)||^3},$$

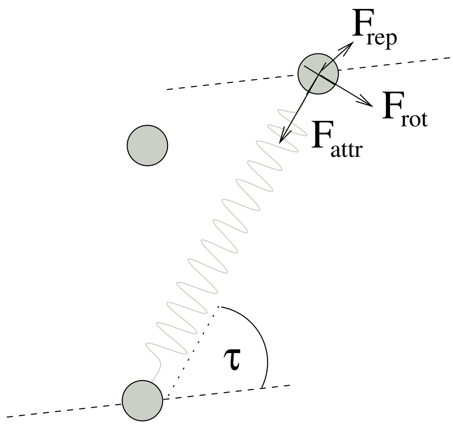

Figure 2: Two nodes connected by a spring. The attractive, repulsive and rotational forces are drawn next to the processed node. The dashed lines show the remembered edge orientation. The attractive and rotational forces are only influenced by nodes directly connected to the current node, whereas the repulsive forces take into account all nodes in the graph.

where γ_{rep} and γ_{attr} are free parameters to adjust the forces. At each step of the spring algorithm, a node v is randomly selected and the force $F(v) = \sum_{w \in V, w \neq v} F(v, w)$ is calculated. Then the node v is moved in the direction of this force. The graph is balanced when the sum of forces between all nodes

$$\sum_{v,w \in V, w \neq v} (F_{\text{rep}}(v, w) + F_{\text{attr}}(v, w))$$

is minimal and therefore the potential energy has reached a minimum. This minimum is not necessarily a global one. To get closer to the global minimum, it is useful to perturb the graph with decreasing amplitude.

The algorithm for a fast and stable solution comes from Fruchterman and Reingold [2] and is already implemented in the graph algorithms library of LEDA [5]. It is a modification of the original spring-embedder model of Eades [1] and works in analogy to forces in natural systems. This algorithm does not solve the edge orientation problem yet, since it only implements spring and repulsive forces. Therefore, Fruchterman's spring algorithm has been modified by adding rotational forces $F_{\text{rot}}(v, w)$ to the springs. These forces take into account the angle information between connected map layer neurons in the neural network (see figure 2).

$$F_{\text{rot}}(v, w) = \gamma_{\text{rot}} \tau(v, w) d(v, w)^2 \delta_{\perp}(v, w)$$

where $\tau(v, w)$ denotes the angle between the current and the stored edge orientation and $\delta_{\perp}(v, w)$ denotes the unit vector normal to $\delta(v, w)$ whose cross product with the preferred edge direction vector does not contain any negative components[1]. Since $\tau(v, w) = \pi - \tau(w, v)$, $F_{\text{rot}}(v, w) = -F_{\text{rot}}(w, v)$.

Both algorithms work for planar as well as for non-planar graphs. The cognitive maps should be mostly planar, however, some few edges can cause non-planarity.

[1] In R^2 there are two possibilities for a unit vector normal to $\delta(v, w)$.

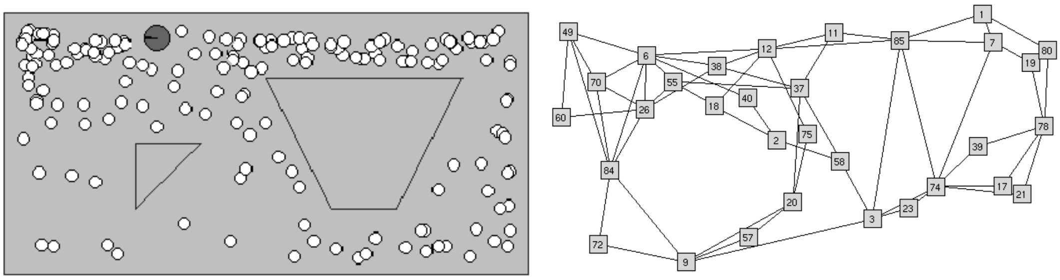

Figure 3: a) The 2D environment fully explored by the robot. The white circles mark the places where the robot has read its distance sensors. Readings were performed every few time steps. b) A cognitive map resulting from the exploration tour of a) after the physical force model has been applied. Each node represents a place cell. One can clearly see the white space as the obstacles. Node 85 has been confused at least once with node 23. Connections between nodes have been drawn when the connection weight was greater than zero.

4 Simulations and Results

For the simulations, an open environment (in contrast to a maze) has been chosen in which the agent can move freely in two dimensions. The whole area is surrounded by a border to keep the agent within the environment it shall explore. In addition, some obstacles which cannot be traversed are placed into the environment (see figure 3). Concerning the neural network parameters, the learning constant γ_r for the weights between input and map layer, the learning constant for the map layer weights γ_α and the maximal value for the map layer connection weights $\alpha_{\max}$ were all set to 0.3. The activation function threshold θ was set to 2.5.

4.1 Sensor Input

Eight distance sensors positioned equidistantly around the robot's body are chosen to receive a very low resolution one-dimensional omnidirectional 'view'. This surround 'view' roughly agrees with the visual field of the rat, which covers 320°. The sensors have a maximal range of 15 times the robot's body size. The sensor reading can be adjusted according to the current bearing of the robot. The choice of distances as a sensor input has not been chosen for its biological plausibility, but rather for its simplicity and for fulfilling a requirement such as continuity, which is essential for the map building.

The action scenario obeys the following rules: the agent explores the environment by moving through it randomly but smoothly. Obstacle avoidance is performed by reading the three front distance sensors and turning accordingly. Every few time steps, the agent transfers its sensor readings to the neural network presented in section 2. These readings include the current compass orientation. Using this method, the cognitive map is gradually generated during exploration. A resulting cognitive map from the environment is depicted in figure 3.

4.2 Analysis of the Place Fields

In figure 4, two typical place field pictures for different numbers of map layer neurons are shown. Noticeable in both pictures is the increased number of place fields around the obstacles. If one viewed these fields as the place fields of an animal, one might explain them as

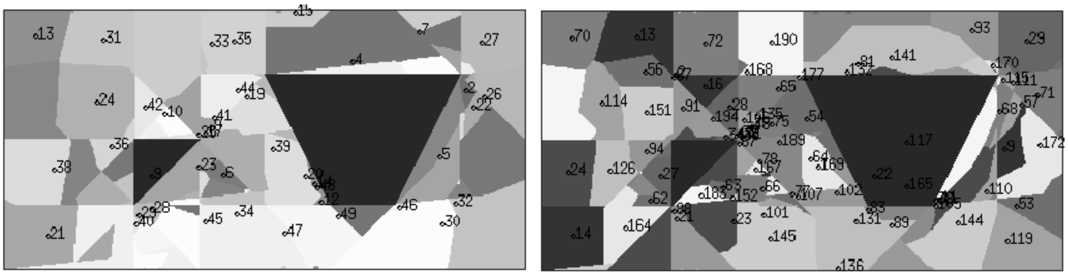

Figure 4: Distribution and shape of the place fields. The grey value of each pixel shows the index of the neuron with the highest activation for sensor measurements taken at this place. Picture a) shows the typical place fields for a neural network with 50 map layer neurons, picture b) uses 200 map layer neurons.

distinctions between more interesting and less interesting places for the animal. The explanation for the situation with the robot is obvious: since the obstacles have convex corners, the distance sensor input changes more frequently and less continuously due to occlusions. It happens to be advantageous for an efficient use of the place cells, since less cells are used for open spaces than for 'interesting' places with many borders.

What happens, if one of the map layer neurons "dies"? The loss of one cell would not cause a large disturbance of the map (due to the absence of the 'Grandmother' cell effect), since the place field would not stay empty, but is most likely subsumed by the cells of neighboring place fields. This occurs because the cell belonging to a particular place field is the cell with the highest activation: other cells might have an only slightly smaller activation in parts of this place field. However, if one map layer cell suddenly disappears, the connectivity of the map is disturbed at this place and has to be relearned locally. The distribution of the winner neuron activities for particular places is shown in figure 5. Each place field can be recognized as being a small activity hill.

Place fields measured in rats look similar to the place fields in these experiments. They are usually a few times the animal's body size, circular, and show firing rates which decrease

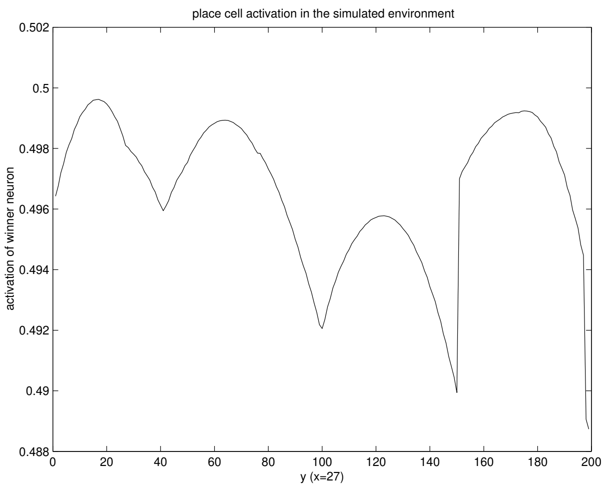

Figure 5: Winner neuron activities on a vertical section through the place fields in the environment. The simulated place fields show Gaussian activity distribution as in rat's place fields.

similarly to Gaussian distributions the further away the rat is from the mass centre of the place field. When the number of map layer neurons is very small, the average place field size is higher than when using more neurons. However, the whole environment is still completely covered. One also notices that with a high number of neurons, some very small place fields are generated, which is not very useful.

5 Discussion

In summary, a computational model of cognitive maps has been created that could be used for navigation tasks in open environments. In several simulated robot experiments it has been shown that a reliable cognitive map can be built during an exploration tour in a virtual environment. Even though the virtual environment did not provide many orientation cues, the sensor input for the robot could be kept very low level. Another advantage is that the cognitive map can be created without the robot getting any external information about its current coordinate position. Metric representations of the initial topological maps have been successfully created by implementing a force model which applied physical forces, as in natural systems, to the place cells in the cognitive map. Many aspects of the presented cognitive map model are biologically plausible and extend existing work.

References

[1] P. Eades. A heuristic for graph drawing. *Congressus Numerantium*, 42:149–160, 1984.

[2] T.M.J. Fruchterman and E.M. Reingold. Graph-drawing by force-directed placement. *Software - Practise and Experience*, 21:1129–1164, 1991.

[3] J. J. Knierim, H. S. Kudrimoti, and B. L. McNaughton. Place cells, head direction cells, and the learning of landmark stability. *Journal of Neuroscience*, 15:1648–1659, 1995.

[4] T. Kohonen. Self-organizing formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69, 1982.

[5] Max-Planck-Institut für Informatik, Saarbrücken. *LEDA, a library of the data types and algorithms of combinatorial computing*.

[6] J. O'Keefe. Place units in the hippocampus of the freely moving rat. *Exp. Neurol.*, 51:78–109, 1976.

[7] J. O'Keefe and A. Speakman. Single unit activity in the rat hippocampus during a spatial memory task. *Experimental Brain Research*, 68:1–27, 1987.

[8] B. Schölkopf and H.A. Mallot. View-Based Cognitive Mapping and Path Planning. *Adaptive Behavior*, 3(3):311–348, 1995.

[9] J. S. Taube, R. U. Muller, and J. B. Ranck. Head-direction cells recorded from the postsubiculum in freely moving rats. *Journal of Neuroscience*, 10:420–447, 1990.

[10] E. C. Tolman. Cognitive maps in rats and men. *Psychological Review*, 55:189–208, 1948.