

Storage of multiple navigation maps using neural networks

Marcelo Trevisan¹, Marco A. P. Idiart¹
and Edson Prestes e Silva Jr.²

¹Instituto de Física,

and

²Instituto de Informática,

Universidade Federal do Rio Grande do Sul,

91501-970 Porto Alegre, RS , Brazil

E-mails: tmarcelo@if.ufrgs.br, idiart@if.ufrgs.br, prestes@inf.ufrgs.br

Abstract

We present a way to store and to recall different environment navigation maps in a neural network. The model is built upon the idea that a navigation map can be written as the solution of the Laplace's problem with suitable boundary conditions applied to obstacles and goals in the environment. The inherent compression of information coming from that allows us to have good storage performances with a reduced number of synaptic connections.

1. Introduction

Some recent works in robotics have focused on the use of harmonic potential functions for path planning [1, 2, 3, 4]. This is an interesting extension of the potential field technique where a potential is built from the superposition of functions representing obstacles and goals. The gradient descent in the resulting potential provides a navigation map.

In the harmonic function method, the solution of the Laplace's equation is used in the place of heuristic potential shapes what brings a series of advantages, the most important being the absence of spurious minima. Therefore the harmonic function method provides a complete algorithm to reach a known goal in a generic environment.

What is important to us in this work is that the computation of the harmonic functions in a grid involves a dynamics where the value of the potential in one site is replaced by the average value of neighboring sites. This resembles an attractor neural network dynamics in a very special network - a place-cells network. Place-cells are pyramidal cells found in the rats hippocampal region that have its firing activity correlated with the position of the animal in the environment. [5, 6, 7, 8, 9]. It is believed that the whole hippocampal region serves as memory device for topographic maps [10, 11, 12]. Therefore we propose a bridge between artificial algorithms and biological models for spatial memory and navigation. In our analogy, however, the neural activities arising from the

harmonic map represent hitting probabilities [4] instead of the subject's position in an absolute frame of reference. In spite of that, we keep calling them place-cells because of its inherent property of having activity correlated with position. We pursue further this idea combining the place-cell network with a context network to store multiple environments.

In what follows we describe the harmonic function method and the complete neural network model that implements it for various environments. Results and perspectives are discussed in the end.

2. Harmonic functions method

A harmonic function on a domain $\Gamma \subset R^d$ is a function which satisfies Laplace's equation

$$\nabla^2 \phi = \sum_{i=1}^d \frac{\partial^2 \phi}{\partial x_i^2} = 0 \quad (1)$$

In the case of robot path planning, the boundary of Γ consists of the boundaries of all obstacles ($\partial\Gamma_o$) and targets ($\partial\Gamma_t$).

In the harmonic functions method the environment is explored and the positions of obstacles and goals are stored in a matrix. Each entry of the matrix represents a point in the environment that is divided in a discrete lattice, or occupancy grid. The Laplace's equation (1) is solved numerically under the constraints that sites with obstacles have fixed values p_o and sites with targets have fixed values p_t , with $p_t < p_o$. The actual values are of no importance. Here for practical purposes we consider $p_o = +1$ and $p_t = -1$.

For a two dimensional environment the numerical calculation can be implemented by the dynamics below, also known as Jacobi's Method,

$$p_{\mathbf{r}}(t+1) = \begin{cases} +1 & \text{if obstacle} \\ -1 & \text{if target} \\ h_{\mathbf{r}}(t) & \text{otherwise} \end{cases} \quad (2)$$

$p_{\mathbf{r}}(t)$ is the place-cell activity (potential) at the grid point \mathbf{r} , and the input function $h_{\mathbf{r}}(t)$ depends on the algorithm we use to solve the Laplace's problem. For nearest

neighbors in a square lattice, the positions are written as $\mathbf{r} = (i, j)$ and we have

$$h_{ij}(t) = \frac{1}{4} \left(p_{i+1,j}(t) + p_{i-1,j}(t) + p_{i,j+1}(t) + p_{i,j-1}(t) \right) \quad (3)$$

If we also consider second neighbors it becomes

$$h_{ij}(t) = \frac{1}{5} \left(p_{i+1,j}(t) + p_{i-1,j}(t) + p_{i,j+1}(t) + p_{i,j-1}(t) \right) + \frac{1}{20} \left(p_{i+1,j+1}(t) + p_{i-1,j+1}(t) + p_{i+1,j-1}(t) + p_{i-1,j-1}(t) \right) \quad (4)$$

Therefore generically we can write

$$h_{\mathbf{r}}(t) = \sum_{\langle \mathbf{r}' \rangle} W_{\mathbf{r}\mathbf{r}'} p_{\mathbf{r}'}(t) \quad (5)$$

where $\langle \mathbf{r}' \rangle$ are sums over neighbors and the weights $W_{\mathbf{r}\mathbf{r}'}$ are hardwired according to the topology of the grid, in such a way that the stable state of the network is a solution of the Laplace equation.

The dynamics (2) clips the obstacle and target activities to fixed values and allows the free space cells (whose position is not occupied neither by an obstacle nor by a target) to relax. The resulting harmonic potential imprinted in the free space cells activities interpolates between the obstacles and the target. Therefore the navigation map is obtained directly from the gradient descent of the activities.

3. The Network Model

There are two distinct neural elements that we need to consider. Place cells, discussed before, represent regions in the environment. They are continuous neurons whose activity, limited between p_o and p_a , depends on the localization of goals and obstacles in the environment. We introduce a second set of neurons - the context neurons. They are binary neurons whose patterns of activity are associated via synaptic learning with specific environment configurations, that are represented by three state patterns

$$\rho_{\mathbf{r}} = \begin{cases} +1 & \text{if there is an obstacle in the site } \mathbf{r} \\ -1 & \text{if there is a goal in the site } \mathbf{r} \\ 0 & \text{for free space cells} \end{cases} \quad (6)$$

We consider a network system composed of three sub-networks with the following properties:

- *Network 1:* It is a feed-forward network that associates environment pattern $\{\rho_{\mathbf{r}}^{\mu}\}_{\mathbf{r} \in \Gamma}$ to context pattern $\{\xi_i^{\mu}\}_{i=1, \dots, M}$.
- *Network 2:* It is a feed-forward network that associates context pattern $\{\xi_i^{\mu}\}_{i=1, \dots, M}$ to environment patterns $\{\rho_{\mathbf{r}}^{\mu}\}_{\mathbf{r} \in \Gamma}$.
- *Network 3:* It is an attractor network with local connections that implements the dynamics in equation (2).

3.1. The network 1 - Context recall

A given environment can be associated to a specific internal state (context) via synaptic strengthening. This association can be obtained through usual supervised learning algorithms. We consider a training set composed of P environment patterns with corresponding context patterns

$$(\{\rho_{\mathbf{r}}^{\mu}\}_{\mathbf{r} \in \Gamma}, \{\xi_a^{\mu}\}_{a=1, \dots, M})_{\mu=1, \dots, P}$$

The ρ patterns are $N \times N$ vectors with components ± 1 or zero. Since we assume that most of the environment is free space, they are rather sparse. The ξ patterns are M -dimensional vectors with components chosen randomly from the distribution

$$P(\xi) = \frac{1}{2} \delta(\xi - 1) + \frac{1}{2} \delta(\xi + 1) \quad (7)$$

The activity of a context neuron is given by

$$s_a = \text{sgn}(h_{\mathbf{r}}) \quad (8)$$

with the local field given by

$$h_a = \sum_{\mathbf{r}} J_{a\mathbf{r}} p_{\mathbf{r}} \quad (9)$$

The synaptic weights are calculated according to an iterative Hebbian method (for instance Rosenblatt's Algorithm) with corrections given by

$$\Delta J_{a\mathbf{r}} = \Theta(-h_a^{\mu} \xi_a^{\mu}) \eta \xi_a^{\mu} \rho_{\mathbf{r}}^{\mu} \quad (10)$$

where $\Theta(x)$ is the Heaviside's function. The network 1 enables a recollection of a context after a partial exploration of a known environment.

3.2. The network 2 - Environment recall

Contexts in their turn can be used as a key to recover the complete environment configuration of obstacles as well as the location of the target. For that an association between context and environment has to be stored. This is the role of the second group of connections that we call network 2. These synapses send information from context cells to environment cell. The activity of an environment cell is given by

$$m_{\mathbf{r}} = f(H_{\mathbf{r}}, \lambda) \quad (11)$$

with the transference function for three states

$$f(x, \lambda) = \Theta(x - \lambda) - \Theta(-x - \lambda)$$

The parameter λ determines when a given input pattern has to be classified as a $+1$, 0 or -1 . Or, in other words, if a given context means an obstacle, free space or a goal in a given position. To keep λ independent of learning the local input has to be normalized, therefore,

$$H_{\mathbf{r}} = \frac{1}{|\mathbf{J}_{\mathbf{r}}|} \sum_a J_{\mathbf{r}a} s_a \quad (12)$$

with

$$|\mathbf{J}_r| = \sum_a (J_{ra})^2$$

The learning procedure suitable for teaching a three states network is somewhat different from the one employed before. We use here an adaptation of the Rosenblatt's algorithm, based in a piecewise linear cost function (see figure 1) [13]. For three-states it reads

$$\Delta J_{ra} = \begin{cases} +\Theta(\tau_i^+ - H_r^\mu) \eta \xi_a^\mu & \text{if } \rho_r^\mu = +1 \\ -\Theta(H_r^\mu + \tau_i^-) \eta \xi_a^\mu & \text{if } \rho_r^\mu = -1 \\ \{ \Theta(-H_r^\mu - \tau_s^-) - \Theta(H_r^\mu - \tau_s^+) \} \eta \xi_a^\mu & \text{if } \rho_r^\mu = 0 \end{cases} \quad (13)$$

where the learning parameters τ define windows where the input field H stay after learning. We consider

$$\begin{aligned} \tau_i^\pm &= \pm(\lambda + \epsilon) \\ \tau_s^\pm &= \pm(\lambda - \epsilon) \end{aligned}$$

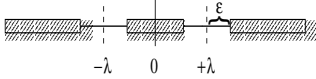


Figure 1: Piecewise linear cost function.

The presence of the parameter ϵ cause learning to be more severe than what is imposed by the training set associated with equation (11). In fact, to learn the training set $\epsilon = 0$ is sufficient. A finite ϵ , however, allows the retrieval properties to be more robust in presence of additional noise. Both λ and ϵ can be optimized, for a given training set and noise, to produce maximal storage.

3.3. The network 3 - Navigation map generation

The place-cell network integrates the environment recall with the generation of the navigation map. A natural way to do that is to consider the following dynamics

$$p_r = g(h_r | H_r, \lambda) \quad (14)$$

where h_r is the local input defined in equation (5) and H_r is the context input defined in equation (12). The transference function is

$$g(x|y, \lambda) = \begin{cases} +1 & \text{if } y > \lambda \\ x & \text{if } y \in [-\lambda, \lambda] \\ -1 & \text{if } y < -\lambda \end{cases} \quad (15)$$

Consequently the stored environment patterns ρ^μ , serve as a mask for the Laplace's dynamics.

Equation (14) together with (8) and (11) compose an attractor system that stores navigation maps for multiple environments. Analytical results for its storage performances will be published elsewhere. Here we are only concerned with the demonstration of its plausibility. For that we implement it for storage of simple environments.

The figure 2 shows the general neural network architecture for a small environment.

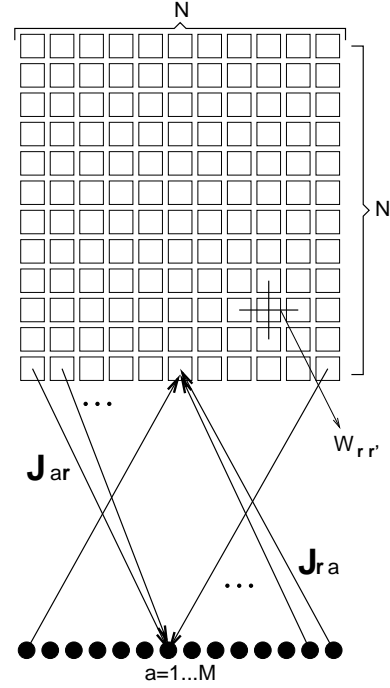


Figure 2: Neural network architecture for a small environment covered by $N \times N$ place cells and a context recall network with M input components.

4. Results

The figures 3 and 4 show the actual environment and the navigation map computed by a neural network. The figures exemplify recall cue by context.

During learning the environment patterns were presented in a cyclic fashion and equation (13) was computed, in period of 30 interactions. In the specific case of figures 3 and 4 three different environments were stored. The learning parameters $\lambda = 0.4$ and $\epsilon = 0.2$ produced the best results for storage.

In 3b and 4b the black (white) squares represent the portions of the environment recalled as obstacles (goals) by the context network. The arrows represent the gradient descent on the harmonic potential calculated by equation (14) using expression (3) after 300 iteration steps.

5. Conclusions

The proposed network is composed of $L = N^2 + M$ neurons. But the number of synaptic weights scales like $\sim N^2(M + 2)$. More importantly the number of modifiable synapses is equal to $\sim N^2M$. This is unusual in attractor neural networks, like Hopfield networks, whose number of synapses is always of order $\sim L^2$. In fact, for storage of generic (random) patterns is always better to used the maximum number of synaptic parameters available. In spite of that, the use of a lower number of synapses does not reduce the network performance here. This is only possible because the patterns that interest us to store are not random. They share an important prop-

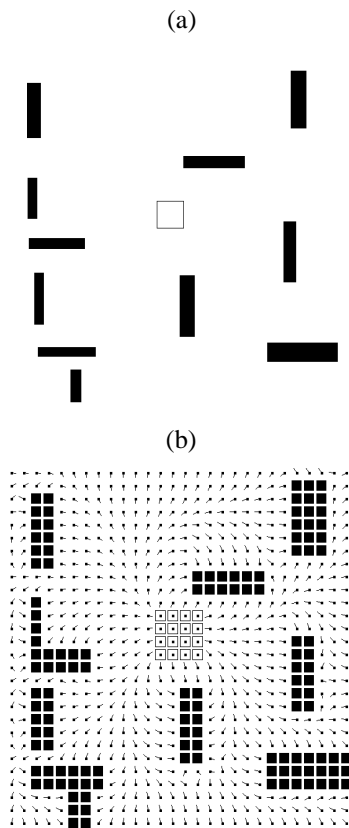


Figure 3: Navigation map for an environment with some obstacles (black rectangles) and a target (white rectangle). (a) The actual environment. (b) Navigation map computed by a neural network with $N \times N = 30 \times 30$ and $M = 64$.

erty – they are the solutions of Laplace’s equation for Dirichlet’s boundary conditions. In fact, the boundary conditions (that describe the environment for us) determine completely the whole place-cell activity pattern therefore only the boundary conditions are needed to be stored. This a nice example of information compression achieved by a physical system.

Acknowledgements This work has been partially supported by Brazilian agencies Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), Financiadora de Estudos e Projetos (FINEP) and Fundação de Amparo à Pesquisa do Rio Grande do Sul (FAPERGS).

References

- [1] C. I. Connolly, J. B. Burns, and R. Weiss. Path planning using Laplace’s equation. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, pages 2102–2106. IEEE, May 1990.
- [2] C. I. Connolly and R. A. Grupen. On the applications of harmonic functions to robotics. *Journal of Robotic Systems*, 10(7):931–946, 1993.
- [3] R. A. Grupen, C. I. Connolly, K. X. Souccar, and W. P. Bursleson. Toward a path co-processor for automated ve-

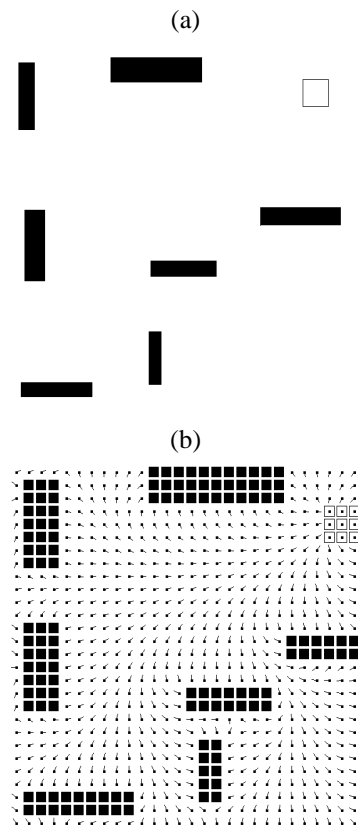


Figure 4: Other obstacles and target configuration. (a) The actual environment. (b) Navigation map computed by the same neural network used before.

hicle control. In *Proceedings of the Intelligent Vehicles ’95 Symposium*. IEEE, 1995.

- [4] C. I. Connolly. Harmonic functions and collision probabilities. *International Journal of Robotic Research*, 16(4):497–507, 1997.
- [5] R. U. Muller, J. L. Kubie, and R. Sayoff. The hippocampus as a cognitive graph. *Hippocampus*, 1(3):243–246, 1991.
- [6] M. A. Wilson and B. L. McNaughton. Dynamics of hippocampal ensemble code for space. *Science, Washington*, 261(5124):1055–1058, 1993.
- [7] B. L. McNaughton, *et al.* Deciphering the hippocampal polyglot: The hippocampus as a path integration system. *Journal of Experimental Biology*, 199(1):173–185, 1996.
- [8] J. O’Keefe and N. Burgess. Geometric determinants of the place fields of hippocampal neurons. *Nature*, 381(6581):425–428, 1996.
- [9] M. R. Metha, C. A. Barnes, and B. L. McNaughton. Experience-dependent, asymmetric expansion of hippocampal place fields. *Proceedings of The National Academy of Sciences USA*, 94:8918–8921, 1997.
- [10] K. I. Blum and L. F. Abbott. A model of spatial map formation in the hippocampus of the rat. *Neural Computation*, 8(1):85–93, 1996.
- [11] L. F. Abbott and K. I. Blum. Functional significance of long-term potentiation for sequence learning and prediction. *Cerebral Cortex*, 6(3):405–416, 1996.
- [12] W. Gerstner and L. F. Abbott. Learning navigational maps through potentiation and modulation of hippocam-

pal place cells. *Journal of Computational Neuroscience*, 4(1):79–94, 1997.

- [13] S. Mertens, H. M. Köhler, and S. Bos. Learning grey-toned patterns in neural networks. *Journal of Physics A: Mathematical and General*, 24(20):4941–4952, 1991.