Topographic RBM as Robot Controller

Paul Hollensen (P)[1], Pitoyo Hartono[2], and Thomas Trappenberg[1]

[1] Dept. of Computer Science, Dalhousie University
[2] Dept. of Mechanics and Information Technology, Chukyo University
E-mail: {hollense,tt}@cs.dal.ca, hartono@sist.chukyo-u.ac.jp

Abstract— In this research we propose learning a controller for a mobile robot with a topographic Restricted Boltzmann machine (tRBM). The topographic RBM generalizes the previously proposed Map-Initialized Perceptron (MIP) to a probabilistic model which learns a joint distribution of sensory states and continuous actions.

Keywords— Topographical Structure, Restricted Boltzmann Machine, Mobile Robot, Imitation Learning

1 Introduction

In previous work we proposed the Map Initialized Perceptron (MIP) [1-2], in which a Self-Organizing Map (SOM) learns a topographic representation of sensory state in an unsupervised manner that are used by a subsequent perceptron which learns appropriate actions to take from a supervisory signal. In our previous work we argued that the use of a SOM to learn an ordered representation has the advantage of being more easily interpretable, aid learning and relearning after injuries, and more closely resembles biological systems. MIP was shown to effectively learn obstacle avoidance on a small mobile robot, e-puck [3].

In this work, we propose a sparse, topographic variation of the restricted Boltzmann machine (tRBM) which generalizes MIP to a probabilistic model where states and actions are modelled jointly. As in [2] we utilize the task of random walking with obstacle avoidance on the e-puck robot to evaluate the model. The topographic representation learned by the tRBM is equally transparent as MIP, allowing us to analyze the functionality acquired during learning. By learning states and actions simultaneously, the tRBM can perform all learning online. Furthermore, the inherently probabilistic nature of the RBM leads it to naturally explore the action space, with unexpected inputs causing more exploratory behavior.

2 RBM as Robot Controller

The Restricted Boltzmann Machine is a stochastic, generative model with symmetric connections between a visible and hidden layer, and no connections within a layer. The model is defined by the energy function

$$E(\mathbf{v},\mathbf{h}) = -\mathbf{b}\cdot\mathbf{v} - \mathbf{c}\cdot\mathbf{h} - \mathbf{v}\cdot\mathbf{W}\cdot\mathbf{h}$$

where **b** and **c** are the visible and hidden biases, respectively, **W** are the weights, and **v** and **h** are the (binary) visible and hidden states. The probability of a visible and hidden node being on are given by

$$\begin{aligned} p(v_i=1|\mathbf{h}) &= \sigma(\mathbf{h}\cdot\mathbf{w}_i^T + b_i) \\ p(h_j=1|\mathbf{v}) &= \sigma(\mathbf{v}\cdot\mathbf{w}_j + c_j) \end{aligned}$$

where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function.

For real-valued data Gaussian visible units are used, in which case (assuming the data is scaled to unit variance for simplicity)

$$\begin{aligned} E(\mathbf{v},\mathbf{h}) &= \sum \tfrac{1}{2}(\mathbf{v}-\mathbf{b})^2 - \mathbf{c}\cdot\mathbf{h} - \mathbf{v}\cdot\mathbf{W}\cdot\mathbf{h} \\ \mathbb{E}[v_i|\mathbf{h}] &= \mathbf{h}\cdot\mathbf{w}_i^T + b_i \end{aligned}$$

The goal of learning is to lower the energy of the data while increasing the energy of everything else, as represented by the equilibrium distribution of the network. The Contrastive Divergence algorithm (CD) [5] efficiently approximates the model distribution with *k*-step reconstructions and works well in practice:

$$\Delta w_{ij} \propto v_i h_j - v_i^r h_i^r$$

where v^r and h^r are the RBM's reconstructions computed by sampling the visibles given the hiddens and vice versa one or more times.

2.1 The Topographic RBM

In order to learn a topographic representation in an RBM, we propose minimizing the difference between the distribution $p(h_{j,k}|\mathbf{v})$ and

$$p(\check{h}_j|\mathbf{v}) = \Sigma_k \mathcal{N}_{j,\sigma}(k) p(h_k|\mathbf{v})$$

where $\mathcal{N}_{j,\sigma}$ is a Gaussian centered on hidden node *j* with variance σ^2. $p(\check{\mathbf{h}}|\mathbf{v})$ can be computed efficiently by convolving $p(\mathbf{h}|\mathbf{v})$ with a small Gaussian filter. For binary hidden units the natural measure of the difference in distributions is the cross entropy, for which the derivative with respect to the weights is simply $(\check{h}_j - h_j)\cdot\mathbf{v}$. Combining this with the CD update yields

$$\Delta w_{ij} \propto v_i h_j - v_i^r h_j^r) + (\check{h}_j - h_j) v_i = v_i \check{h}_j - v_i^r h_j^r$$

2.2 Sparsity Regulation

While SOMs offer a very local representation due to its winning node learning mechanism, a standard RBM learns a very distributed representation. Much previous work has shown the advantage of sparse representations, which lie between these two, in a variety of tasks, as well as their resemblance to neuronal representations [6].

A target sparsity level, ρ, can be enforced on individual nodes by maintaining an estimate of their expected activation, $q_j(t) = (1-\lambda)q_j(t-1) + \lambda h_j(t)$, and again minimizing the cross entropy between the distributions $p(h_j = 1) \approx q$ and $p(h_j = 1) = \rho$ [5]. In this case, $\Delta w_{ij} \propto v_i(h_j^t + \rho - q_j) - v_i^r h_j^r$. This has the added advantage of preventing *dead nodes*, forcing the network to find a representation which utilizes all hidden nodes.

We propose an alternative mechanism in which global hidden activity triggers lateral inhibition when it exceeds the sparsity target. In this case, $p(\mathbf{h}^s) = p(\mathbf{h}) + \min(0, \rho - \mu_{p(\mathbf{h})})$. While this would be vulnerable to the dead node problem in a standard RBM, the neighbor excitation in the topographic RBM ensures that activity, and thus learning, is spread through the network. Both methods were effective in regulating activation sparsity and resulted in equivalent performance on the experimental task. However, the global mechanism more closely resembles neuronal systems and does not require any memory of activity levels.

2.3 Jointly modelling states and actions

Rather than modelling state with a SOM or other network and then modelling actions on top of this as in MIP, we can jointly model sensory states and actions with the RBM. When actions are supplied by a supervisor, which in this case is a hard-coded algorithm but could just as easily be a human, the RBM performs *imitation learning*. When the supervisory signal is missing we need to compute the expected action given the sensory state rather than hidden state. To do this, Gibbs sampling is used, where the action nodes are initialized at an arbitrary value (e.g. 0) and then hidden states are samples and action states reconstructed one or more times

$$\begin{aligned} p(h_j^n = 1|\mathbf{s}, \mathbf{a}^n) &= \sigma(\mathbf{s} \cdot \mathbf{w}_j + \mathbf{a}^n \cdot \mathbf{u}_j + c_j) \\ a_k^{n+1} &= \mathbf{h}^n \cdot \mathbf{u}_k + b_k^a \end{aligned}$$

where $\mathbf{w}$ and $\mathbf{u}$ connect the sensory inputs and actions, respectively, to the hidden layer, and $\mathbf{b}^a$ are the action biases.

2.4 Regulating exploration with Temperature

We can paramaterize the exploration-exploitation tendency of the robot by introducing a variable temperature, *T*, to the system so that

$$p(h_j^n = 1|\mathbf{s}, \mathbf{a}^n) = \sigma\left(T^{-1}(\mathbf{s} \cdot \mathbf{w}_j + \mathbf{a}^n \cdot \mathbf{u}_j + c_j)\right)$$

When the temperature is high, the input to the hidden layer is scaled down resulting in greater stochasticity in the hidden layer activity, and thus more exploratory behavior. By scaling the temperature down over learning, known as simulated annealing, we can gradually exploit the acquired knowledge more and more.

3 Experiments

In the experiment, the network observes the proximity sensor readings and the actions selected by the hand-coded obstacle avoidance algorithm. As seen in figure 1, the network learns distinct topographic representations for the states corresponding to the behaviors going forward, backward, soft left, hard left, soft right, and hard right. We also see that more hidden activity is devoted to the situations which deviate more from the default state of sensing no obstacles and going forward. This default state can be entirely modelled with the visible biases, and thus causes no hidden activity.

As depicted by the error plot in figure 3, the topographic RBM successfully learns the mapping from proximity sensor state to obstacle avoiding actions. The early spike in error occurs when sparsity regulation is enabled, however the network quickly recovers. Sparsity regulation is delayed in order to allow the network to form a tightly interwoven topography.

4 Conclusion and future work

We have shown an RBM model which learns sparse, topographic representations that successfully model the joint distribution of sensory and action states in a mobile robot executing an obstacle avoidance behavior. The topographic structure makes the knowledge acquired by the network interpretable and relatable to biological systems while the probabilistic nature of the model addresses the exploration-exploitation tradeoff.

RBMs offer a number of features which enable extensions to the present work. By stacking RBMs in a Deep Belief Network (DBN), hierarchical representations of high-dimensional inputs such as visual imagery can be modelled and compressed before being fed into a top-level RBM jointly modelling states and actions. Furthermore, the paradigm of free energy based reinforcement learning can be integrated to allow the supervisory signal to be replaced with a reward signal from the environment.

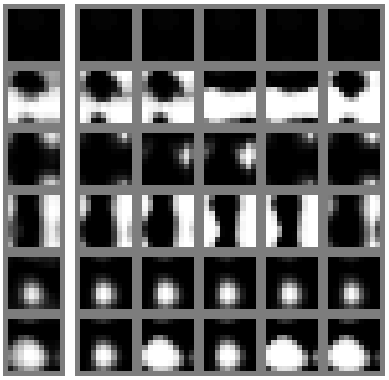

Figure 1: Mean (far left) and random sample hidden activation when going forward (row 1), backward (2), soft left (3), hard left (4), soft right (5), hard right (6).

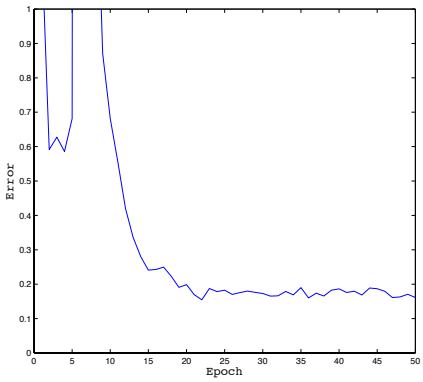

Figure 2: Error in real-valued action selection for obstacle avoidance task.

References

[1] Trappenberg, T., Saito, A., & Hartono, P. (2010). Selective attention improves self-organization of cortical maps with multiple inputs. *IJCNN 2010*, 1–4.

[2] P. Hartono and T. Trappenberg, Internal Topographical Structure in Training Autonomous Robot, Proc. IEEE SMC 2011 (accepted).

[3] Mondada, F., et al. (2009). The e-puck, A robot designed for education in engineering. *9th Conf. on Autonomous Robot Systems and Competitions*, 1, 59–65.

[4] Hinton, G.E. (2002). Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14, 1711–1800.

[5] Hinton, G. (2010). A Practical Guide to Training Restricted Boltzmann Machines. *U. of Toronto Technical Report UTML TR 2010-003*.

[6] Foldiak, P., Endres, D. (2008). Sparse coding. *Scholarpedia*, 3, 2984.