Network Capacity for Latent Attractor Computation *

Simona Doboli
Complex Adaptive Systems Laboratory
ECECS Department
University of Cincinnati
Cincinnati, OH 45221-0030
(sdoboli@ececs.uc.edu)

Ali A. Minai
Complex Adaptive Systems Laboratory
ECECS Department
University of Cincinnati
Cincinnati, OH 45221-0030
(Ali.Minai@uc.edu)

Abstract

Attractor networks have been one of the most successful paradigms in neural computation, and have been used as models of computation in the nervous system. Many experimentally observed phenomena — such as coherent population codes, contextual representations, and replay of learned neural activity patterns — are explained well by attractor dynamics. Recently, we proposed a paradigm called "latent attractors" where attractors embedded in a recurrent network via Hebbian learning are used to *channel* network response to external input rather than becoming manifest themselves. This allows the network to generate context-sensitive internal codes in complex situations. Latent attractors are particularly helpful in explaining computations within the hippocampus — a brain region of fundamental significance for memory and spatial learning. The performance of latent attractor networks depends on the number of such attractors that a network can sustain. Following methods developed for associative memory networks, we present analytical and computational results on the capacity of latent attractor networks.

Introduction

Attractor neural networks have become a dominant paradigm in neural computation. While such networks have primarily been used as associative memories [10, 8], they have also been used to model several other phenomena such as the stabilization of population codes [12]. Attractor networks have been especially useful for modeling processes within the hippocampal region of the mammalian brain, since this region is known to have the strong recurrent connectivity necessary for attractor dynamics. There are attractor-based models of associative memory storage and recall [13, 14, 11, 17, 9], spatial representations [20, 18, 16], context-dependent coding [18, 15, 5, 6, 4], path-integration [14, 18, 16], and route learning [2, 19] in the hippocampus.

Recently, we have proposed a paradigm called *latent attractor computation*, which is useful for generating context-dependent codes or responses to external inputs. The basic idea is to use weak attractors for the modulation of information flow in the primary data path (which may be feed-forward or recurrent). The attractors are too weak to sustain themselves — hence the term "latent attractor" — but the application of external input "unmasks" them, channeling the network's response in a metastable way. Thus, the latent attractors act as dynamically reconfigurable, metastable biases on the primary signals, producing a long-term dependency on initial inputs without disrupting the system's ability to encode new information (see [15, 5, 6, 4] for more details).

*Proc. IJCNN'2000, Como, Italy, pp. 222-228

System Description

While latent attractors can be embodied in many architectures, here we study the system shown in Figure 1. The network has two processing layers and a stimulus layer. The *stimulus layer*, $\mathcal{S}$, has N_S binary neurons, of which K_S are active at a time. Stimulus patterns are drawn randomly from a *stimulus set*, $\mathcal{I}$. The *response layer*, $\mathcal{R}$, has N_R binary neurons with K_R active at a time, and the *hidden layer*, $\mathcal{H}$, has N_H binary neurons with K_H active. The connections from $\mathcal{S}$ to $\mathcal{R}$ are random, with probability of connection C_S. Active connections are set to weight W_S.

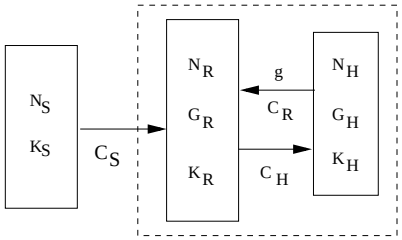

Figure 1: System architecture

The latent attractors are embedded into the $\mathcal{R}$ and $\mathcal{H}$ layers. Each attractor, α, comprises two patterns, one in each of the layers $\mathcal{R}$ and $\mathcal{H}$. The layer $\mathcal{R}$ pattern, $R^\alpha = \{r_1^\alpha r_2^\alpha ... r_{N_R}^\alpha\}$, has $G_R > K_R$ active neurons — called the *active set* of α, and denoted A_R^α — and $N_R - G_R$ inactive ones. Similarly, the layer $\mathcal{H}$ pattern, $H^\alpha = \{h_1^\alpha h_2^\alpha ... h_{N_R}^\alpha\}$, has $G_H > K_H$ active neurons (set A_H^α). A total of M attractors are generated randomly, each with the same activity. Note that only a fraction of neurons in an attractor's active set can actually be active at a time.

The $\mathcal{R}$ and $\mathcal{H}$ layers project to one another, with the weights set as follows. First, the $\mathcal{H} \to \mathcal{R}$ and $\mathcal{R} \to \mathcal{H}$ connections are selected randomly with probabilities of connection C_R, and C_H, respectively. The weight, w_{ij}, from neuron $j \in \mathcal{H}$ to $i \in \mathcal{R}$ is set as $w_{ij} = c_{ij}\Theta\left(\sum_\alpha r_i^\alpha h_j^\alpha\right)$, where c_{ij} is a binary variable indicating the existence of the j-to-i connection, and $\Theta(x) = 1$ if $x > 0$ and 0 else. Similarly, the weights from $i \in \mathcal{R}$ neurons to $j \in \mathcal{H}$ neurons are set. This is the Hebbian rule first studied by Willshaw et al. [21]. Once the weights are set, the M pairs of patterns are stored as attractors, or fixed points in the 2-layer network.

The dendritic sums for neurons $i \in \mathcal{R}$ and $j \in \mathcal{H}$ at time t are:

$$h_i(t) = g\sum_{j=1}^{N_H} w_{ij}x_j(t-1) + \sum_{k=1}^{N_S} w_{ik}x_k(t), \quad h_j(t) = \sum_{i=1}^{N_R} w_{ji}x_i(t) \tag{1}$$

where g is the gain of the connection from $\mathcal{H}$ to $\mathcal{R}$. The K_R and K_H cells with the highest dendritic sums are fired in $\mathcal{R}$ and $\mathcal{H}$, respectively.

The stimulus patterns for the functioning system are seen as falling into two classes: 1) *Context inputs*, which preferentially (though not exclusively) stimulate the active set of one specific attractor — termed the *target attractor*; and 2) *Regular inputs*, which stimulate neurons in all attractors with equal probability. When a context pattern stimulates the system, neurons in the target attractor's active set are over-represented in the group of K_R and K_H active neurons. This, in turn, gives the active set of the target attractor an edge in responding to subsequent regular stimuli, and activity stays confined within this set until another context pattern appears and causes an attractor switch. The key point is that, while the network response to regular stimuli is confined within the target attractor's active set, the actual pattern of activity *within the attractor* is determined by the external regular stimulus. Thus, the response is conditioned by the last context pattern seen by the system — however remote in time — while preserving the information present in the current regular stimulus [15, 5, 6, 4].

System Analysis

Clearly, the functioning of this system requires that only context patterns be allowed to cause attractor switches. Regular patterns, which stimulate all attractors randomly, should not lead to switching. The analysis in this paper has two aims: 1) to determine the capacity of the system, M_{max}, i.e., the maximum number of latent attractors that can be stabilized against spurious switching; and 2) To predict the dynamical evolution of the network in terms of attractor stability. The standard signal-to-noise approach is used, [7, 3], albeit taking account of the correlations between weights. This leads to dynamical equations describing the evolution of network state, which are used to estimate capacity numerically.

The dendritic sum for neuron $i \in \mathcal{R}$ has distinct distributions depending on whether i is in the active set of the currently selected attractor. For convenience, we assume throughout that attractor $\alpha = 1$ is the selected one. For $i \in \mathcal{R}$, the dendritic sum can be written as follows:

$$h_i(t) = g \sum_{j \in S_H^1} w_{ij} x_j(t-1) + g \sum_{j \notin S_H^1} w_{ij} x_j(t-1) + \sum_{k=1}^{N_S} w_{ik} x_k(t) \quad for\ i \in A_R^1, \tag{2}$$
$$h_i(t) = g \sum_{j \in S_H^1} w_{ij} x_j(t-1) + g \sum_{j \notin S_H^1} w_{ij} x_j(t-1) + \sum_{k=1}^{N_S} w_{ik} x_k(t) \quad for\ i \notin A_R^1.$$

The same applies to the dendritic sums of neurons $j \in \mathcal{H}$:

$$h_j(t) = \sum_{i \in S_R^1} w_{ji} x_i(t) + \sum_{i \notin S_R^1} w_{ji} x_i(t)\ for\ j \in A_H^1, \quad h_j(t) = \sum_{i \in S_R^1} w_{ji} x_i(t) + \sum_{i \notin S_R^1} w_{ji} x_i(t)\ for\ j \notin A_H^1 \tag{3}$$

The sums for $i, j \in A_{R/H}^1$ are designated *high*, and those for $i, j \notin A_{R/H}^1$ are termed *low*. If the number of active neurons in the attractor's active set ($n_{gR}(t)$, and $n_{gH}(t-1)$) and outside it ($n_{sR}(t)$, and $n_{sH}(t-1)$), and also the number of active neurons in the $\mathcal{S}$ layer ($n_S(t) = K_S$) are known, the sums in the previous relations become sums of binary variables, the weights, that depend on the weight distributions: $P(w_{ij} = 1)$, $P(w_{ji} = 1)$, and $P(w_{ik} = W_S)$. In the limit, when the $n_{(.)}(t)$ are very large, the sums can be approximated with normal distributions completely defined by mean and variance.

For weights between $\mathcal{S}$ and $\mathcal{R}$, $P(w_{ik} = W_S) = C_S$. For the weights between $\mathcal{R}$ and $\mathcal{H}$, two cases arise depending on whether the source and destination neurons are part or not of the current target attractor's active set. If both source and destination neurons are part of attractor 1's active set, then $P(w_{ij} = 1 | i \in A_R^1, j \in A_H^1) = C_R$ and $P(w_{ji} = 1 | i \in A_R^1, j \in A_H^1) = C_H$. In the other three situations, where either source or destination neurons, or both do not belong to attractor 1, the probability that a weight is modified is given by [3, 7]: $\rho_R = P(w_{ij} = 1 | i \notin A_R^1 \vee j \notin A_H^1) = C_R[1 - (1 - a_R a_H)^{(M-1)}]$, where $a_R = G_R/N_R$ and $a_H = G_H/N_H$. Similarly, for $\mathcal{R}$-to-$\mathcal{H}$ weights, $\rho_H = C_H[1 - (1 - a_R a_H)^{(M-1)}]$. The mean of the dendritic sums are then:

$$\begin{aligned} \mu_i(t) &= g n_{gH}(t-1) C_R + g n_{sH}(t-1) \rho_R + W_S K_S C_S \quad for\ i \in A_R^1, \\ \mu_i(t) &= g n_{gH}(t-1) \rho_R + g n_{sH}(t-1) \rho_R + W_S K_S C_S \quad for\ i \notin A_R^1, \end{aligned} \tag{4}$$

for R neurons, and for H neurons:

$$\mu_j(t) = n_{gR}(t) C_H + n_{sR}(t) \rho_H \quad for\ i \in A_H^1, \quad \mu_j(t) = n_{gR}(t) \rho_H + n_{sR}(t) \rho_H \quad for\ i \notin A_H^1 \tag{5}$$

In evaluating the variance of the dendritic sums, several different levels of approximation can be used, depending on whether the elements of the sums (relations (2), (3)) are considered independent or not. The simplest case would be to neglect all dependencies that build up between weights due to shared patterns (Level 0 model, [7]), but this gives a poor match with simulation results for both, capacity, and dynamics. The next possibility is to consider the correlations that form between weights (Level 1 model [7]), which leads to the following results for the variance of dendritic sums:

$$\begin{aligned} &Var[h_i(t)] = \\ &\quad g^2 n_{gH}(t-1) C_R (1 - C_R) + g^2 n_{sH}(t-1) \rho_R (1 - \rho_R) + g^2 n_{sH}(t-1)^2 \gamma_R + W_S K_S C_S (1 - C_S) \quad for\ i \in A_R^1, \\ &Var[h_i(t)] = \\ &\quad g^2 [n_{gH}(t-1) + n_{sH}(t-1)] \rho_R (1 - \rho_R) + g^2 [n_{gH}(t-1) + n_{sH}(t-1)]^2 \gamma_R + W_S K_S C_S (1 - C_S) \quad for\ i \notin A_R^1 \end{aligned} \tag{6}$$

for $i \in \mathcal{R}$, and:

$$\begin{aligned} Var[h_j(t)] &= n_{gR}(t)C_H(1-C_H) + n_{sR}(t)\rho_H(1-\rho_H) + n_{sR}(t)^2\gamma_H \quad for\ j \in A_H^1, \\ Var[h_j(t)] &= [n_{gR}(t) + n_{sR}(t)]\rho_H(1-\rho_H) + [n_{gR}(t) + n_{sR}(t)]^2\gamma_H \quad for\ j \notin A_H^1, \end{aligned} \tag{7}$$

for $j \in \mathcal{H}$, where $\gamma_{(.)}$ represent the covariance between two weights:

$$\gamma_R = Cov[w_{ij_1}, w_{ij_2} | i \notin A_R^1 \vee j_1, j_2 \notin A_H^1] = C_R^2[(1 - 2a_R a_H + a_R a_H^2)^M - (1 - a_R a_H)^{2M}] \tag{8}$$

For γ_H a similar computation is used, with the only difference in the change of a_R with a_H.

Using expressions (4), (5) for mean and (6), (7) for variance of the dendritic sums, a threshold that will fire approximately the K most active neurons can be computed in an iterative way as described below. The algorithm receives as input the number of correctly and erroneous firings at time $t-1$ and it outputs both the threshold and the updated values for the number of firings at step (t). These are first evaluated for the $\mathcal{R}$ layer, and then for the $\mathcal{H}$ layer, with the input values taken from the output of the $\mathcal{R}$ layer. The algorithm first computes a high dendritic sum threshold such that $K_{(.)}$ neurons in $A_{(.)}^1$ are above it, and a low dendritic sum threshold such that 1 neuron outside $A_{(.)}^1$ is above threshold (0 would result in an infinite threshold under the Gaussian approximation). If the high threshold exceeds the low one, at least $K_{(.)} - 1$ neurons in the selected attractor can be fired with at most one spurious firing. Otherwise, the high and low thresholds are increased and decreased, respectively, keeping the sum of suprathreshold neurons fixed at $K_{(.)} + 1$, until the high threshold is greater than or equal to the low one. This iterative algorithm for computing the threshold and updating the number of genuine and spurious firings allows us to simulate the effect of the K-of-N competitive firing, albeit not in closed form.

To estimate the capacity, we start by choosing the number of attractors, $M = 1$. The equations (4) - (8) are initialized with $n_{gR} = K_R$ and $n_{sR} = 0$ (all initial activity within the active set of the target attractor), and iterated for 100 steps, using the threshold estimation described above to determine n_{gR} and n_{sR} at each step. Stability is estimated by computing the following expression over steps 91 to 100:

$$L_R = \frac{G_R}{K_R}\left(\frac{< n_{gR} >}{G_R} - \frac{< n_{sR} >}{N_R - G_R}\right), \tag{9}$$

where $< . >$ denotes time-averaging. L_R is a measure of stable confinement of activity within the target attractor as a function of network size and attractor activity. If $L_R < 0.95$, the configuration is declared unstable. Otherwise, the number of attractors is increased by one, and the procedure restarted. This continues until the $L_R < 0.95$ condition is satisfied; the M value at that point is the capacity estimate.

The capacity estimates derived from iterating the equations were tested via network simulations. The independent parameter for the simulations was $a = a_R = G_R/N_R = a_H = G_H/N_H$ — the ratio between the active set of the attractors and total number of neurons in each layer. The recurrent gain g was set to equalize the mean of the recurrent input and the mean of the external stimulus for layer $\mathcal{R}$ neurons in the active set if all neurons that fired at the previous step were also within the set: $g = W_S K_S C_S/(K_H C_R)$.

Results and Discussion

The results show that estimates of latent attractor capacity are very close to the experimental values, with bigger errors in regions where the Gaussian approximation is not very good: small a and/or sparse activity (K_R, K_H small). However, in the limit of very large networks, the theoretical estimates match the simulation results very well.

The graph in Figure 2(a) shows the empirical and estimated capacities for a network with $N_S = 400$, $N_R = 2000$, $N_H = 500$, and an activity level (K/G) of 20% in the $\mathcal{R}$ layer and 90% in the $\mathcal{H}$ layer. The estimates are excellent for $a > 0.15$ but smaller a values produce some error. To determine the effect of network size, the number of neurons in each layer was doubled with the results presented in Figure 2(b). As expected, the error between estimation and simulation decreased in the region of small a's.

In the next simulation (Figure 2(c)) the activity with respect to the size of the active set was increased : $K_R = 0.6G_R$, while the network dimension was kept at the values used in (a). In this case, the estimates are much closer to the simulation results even for very small a.

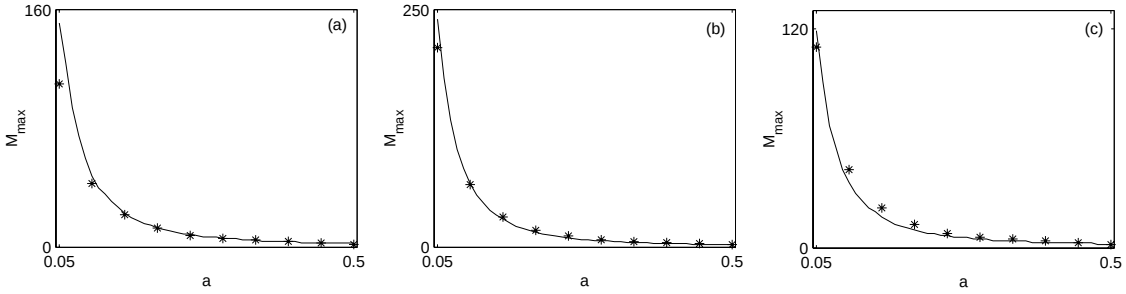

Figure 2: Capacity results (estimation - solid line), (simulation - $\star$): (a) $N_S = 400$, $N_R = 2000$, $N_H = 500$, $K_R = 0.2G_R$; (b) $N_S = 800$, $N_R = 4000$, $N_H = 1000$, $K_R = 0.2G_R$; (c) $N_S = 400$, $N_R = 2000$, $N_H = 500$, $K_R = 0.6G_R$.

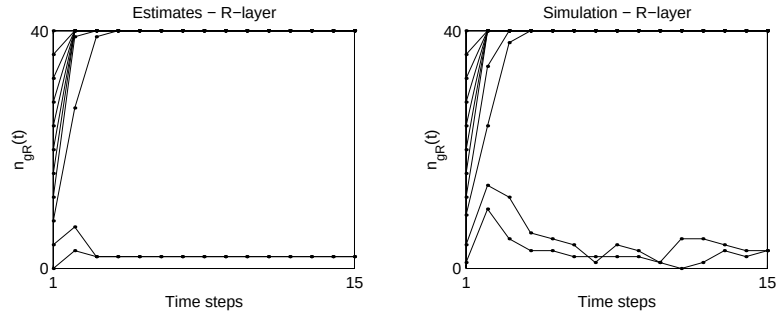

Figure 3: Network dynamics: $N_S = 400$, $N_R = 2000$, $N_H = 500$, $K_R = 0.2G_R$

The analysis presented above works well for capacity estimation because the initial state is in the chosen attractor. When a significant part of the neurons are activated outside the attractor's active set (i.e., $n_{gR}/K_R < 1$, the correlations that build up in time between the current state and the previous ones become important, and the previously derived relations for mean and variance are not longer accurate. In order to estimate the dynamical evolution of the network state in such situations, a more detailed analysis was done taking the dependence of current state on previous state into account [1, 7]. The theoretical and simulation results of the evolution of network state overlap with the selected attractor are presented in Figure 3. The estimates show that after a certain initial distance from the selected attractor, the activity does no longer converge towards that attractor. They are in line with the results of network simulation.

Acknowledgement: This research was supported by NSF Grant No. IBN-9808664 and the University Research Council, University of Cincinnati.

References

[1] S. Amari and K. Maginu. Statistical neurodynamics of associative memory. *Neural Networks*, 1:63–73, 1988.

[2] K.I. Blum and L.F. Abbott. A model of spatial map formation in the hippocampus of the rat. *Neural Comput.*, 8:85–93, 1996.

[3] J. Buckingham and Willshaw D. Performance characteristics of the associative net. *Network*, 3:407–414, 1992.

[4] S. Doboli, A.A. Minai, and P.J. Best. Generating smooth context-dependent representations. In *Proc. of IJCNN'99, Washington D.C.*, 1999.

[5] S. Doboli, A.A. Minai, and P.J. Best. A latent attractors model of context selection in the dentate gyrus-hilus system. *Neurocomputing*, 26-27:671–676, 1999.

[6] S. Doboli, A.A. Minai, and P.J. Best. Latent attractors: a model for context-dependence place representations in the hippocampus. *Neural Computation*, 12(5), 2000.

[7] W.G. Gibson and Robinson J. Statistical analysis of the dynamics of a sparse associative memory. *Neural Networks*, 5:645–661, 1992.

[8] S. Grossberg. *Neural Networks and Natural Intelligence*. Cambridge, MA: MIT Press, 1988.

[9] M.E. Hasselmo, E. Schnell, and E. Barkai. Dynamics of learning and recall at excitatory recurrent synapses and cholinergic modulation in hippocampal region CA3. *J. Neurosci.*, 15:5249–5262, 1995.

[10] J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proc. Nat. Acad. Sci. USA*, 79:2554–2558, 1982.

[11] W.B. Levy. A computational approach to hippocampal function. In R.D. Hawkins and G.H. Bower, editors, *Computational Models of Learning in Simple Neural Systems*, pages 243–305. Academic Press, San Diego, CA, 1989.

[12] A.V. Lukashin and A.P. Georgopoulos. A dynamical neural network model for motor cortical activity during movement: population coding of movement trajectories. *Biological Cybernetics*, 69(5/6):517–524, 1993.

[13] D. Marr. Simple memory: A theory for archicortex. *Phil. Trans. R. Soc. Lond. B*, 262:23–81, 1971.

[14] B.L. McNaughton and R.G.M. Morris. Hippocampal synaptic enhancement and storage within a distributed memory system. *Trends in Neurosci.*, 10:408–415, 1987.

[15] A.A. Minai and P.J. Best. Encoding spatial context: A hypothesis on the function of the dentate gyrus-hilus system. In *Proc. Int. Joint Conf. on Neural Networks, Anchorage*, pages 587–592, 1998.

[16] A.D. Redish and D.S. Touretzky. The role of the hippocampus in solving the Morris water maze. *Neural Comp.*, 10:73–111, 1998.

[17] E.T. Rolls. The representation and storage of information in neuronal networks in the primate cerebral cortex and hippocampus. In R. Durbin, C. Miall, and G. Mitchison, editors, *The Computing Neuron*, pages 125–159. Addison-Wesley, Reading, MA, 1989.

[18] A. Samsonovich and B.L. McNaughton. Path integration and cognitive mapping in a continuous attractor neural network model. *J. Neurosci.*, 17:5900–5920, 1997.

[19] W.E. Skaggs and B.L. McNaughton. Spatial firing properties of hippocampal CA1 populations in an environment containing two visually identical regions. *J. Neurosci.*, 15:811–820, 1998.

[20] M. Tsodyks and T Sejnowski. Associative memory and hippocampal place cells. *Int. Journal of Neural Systems*, supp. 1995:81–86, 1995.

[21] D. Willshaw, O.P. Buneman, and H.C. Longuet-Higgins. Non-holographic associative memory. *Nature*, 222:960–962, 1969.