Static and Dynamic Attractors of Autoassociative Neural Networks

Dmitry O. Gorodnichy[1] and Alexandre M. Reznik[2]

[1] Dept. of Computing Science, University of Alberta, Edmonton, Alberta, Canada T6G 2H1
[2] IPMMS, Cybernetics Center of Ukrainian Ac.Sc., Kiev, Ukraine

Abstract. In this paper we study the problem of the occurrence of cycles in autoassociative neural networks. We call these cycles *dynamic attractors*, show when and why they occur and how they can be identified. Of particular interest is the pseudo-inverse network with reduced self-connection. We prove that it has dynamic attractors, which occur with a probability proportional to the number of prototypes and the degree of weight reduction. We show how to predict and avoid them.

Keywords: pattern recognition, neural network, pseudo-inverse rule, stable state.

1 Introduction

Autoassociative neural networks, like those introduced by Amari [1], Kohonen [2], Hopfield [3], Personnez [4], are intensively used for pattern recognition and low-level computer vision problems [5, 6, 7]. These problems include identification and categorization of faces, computation of optical flow, static and motion stereo, image restoration, and other ill-posed according to Hadamard [8] problems. What makes these networks attractive is 1) their cooperative computation, which provides not only massive parallelism but also a great degree of robustness, 2) analogy to biological nervous systems and 3) fast learning and evaluation.

The self-organizing feature of these networks is attributed to their ability to converge from an arbitrary state to a *stable state*, which we will call a *static attractor*. A state can be a vector of pixel intensities or a vector of facial features, with faces to be "stored" as static attractors, as in face recognition [9], or it can be a vector of possible values of the unknowns of a problem, with attractors corresponding to the solutions of this problem, as in computer vision [6]. The number of static attractors determines the *capacity* of the network, which is the number of prototypes a network of size N is able to recognize.

It is known that when the number of attractors exceeds a certain boundary, the network may loose the ability to converge to desired attractors. For the networks learnt by the Hebbian rule (as in [3]) this boundary is $0.14N$ and for the network learnt by the pseudo-inverse learning rule (as in [4]) it is $0.5N$. If other learning rules are used, the network may converge not to a *stable state* but to a *cycle* of a number of states. These cycles are referred to as *dynamic attractors*. More specifically, cycles may occur when the network with non-zero self-connection ($C_{ii} \neq 0$) evolves according to the *synchronous update rule* [10, 11]. This is the case for the networks Zhou and Chelappa [6] use for computer vision problems and this is also the case for the network suggested by Gorodnichy [12], which exhibits an improved capacity of up to $0.75N$ and which is of interest in this paper.

It is important to be aware of the dynamic attractors so that they can be identified and avoided.

In this paper, we show first when and why dynamic attractors occur in autoassociative networks. Then we focus our attention on the network considered in [12, 13], which is the pseudo-inverse network with reduced self-connection. We prove that this network has dynamic attractors and show how to predict and avoid them.

2 The Nature of Attractors

2.1 The Model

The network consists of N mutually interconnected two-state neurons $y_i : y_i \in \{+1, -1\}$. The evolution of the network in time is determined by a synchronous *update rule*

$$y_i(t+1) = \text{sgn}[s_i(t) - b_i] = \begin{cases} s_i(t) - b_i, & \text{if } s_i(t) > b_i \\ -(s_i(t) - b_i), & \text{otherwise} \end{cases} \tag{1}$$

where $s_i(t) = \sum_{j=1}^{N} C_{ij} y_j(t)$ is a *postconnection potential* and b_i is a threshold of a neuron. C_{ij} are *interconnection weights*, which are assumed to be symmetric, i.e. $C_{ij} = C_{ji}$, with self-connection C_{ii} not necessarily being zero.

In vector form the update rule can be rewritten as

$$\mathbf{Y}(t+1) = \text{sgn}[\mathbf{S}(t) - \mathbf{B}], \quad \mathbf{S}(t) = \mathbf{C}\mathbf{Y}(t). \tag{2}$$

Vector $\mathbf{Y}(t)$ is referred to as a *state of the network* and matrix $\mathbf{C}$: NxN is a *weight matrix*. The weight matrix is calculated from a requirement that a set of prototypes $\mathbf{V}^1, ..., \mathbf{V}^M$ be static attractors of the network. One way of achieving this is to use the Hebbian, outer product, *learning rule*: $\mathbf{C} = \frac{1}{N}\mathbf{V}\mathbf{V}^T$, where $\mathbf{V}$ is the matrix made of column prototype vectors.

In this paper, in Section 3, we consider another rule, called the *pseudo-inverse learning rule*, which exhibits much better information-retrieval capability than the Hebbian one [1, 4], and which is defined as $\mathbf{C} = \mathbf{V}\mathbf{V}^+$, where $\mathbf{V}^+$ is the *pseudoinverse* of matrix $\mathbf{V}$. The iterative formula[1] of this rule is

$$C_{ij}^m = C_{ij}^{m-1} + \frac{(v_i^m - s_i^m)(v_j^m - s_j^m)}{E^2}, \quad \text{if } E^2 > 0, \tag{3}$$

[1] See [14] for the derivation of the formula.

where $E^2 \doteq \|\mathbf{C}^{m-1}\mathbf{V}^m - \mathbf{V}^m\|^2 = N - \sum_{i=1}^N v_i^m s_i^m$ and $s_k^m = \sum_{i=1}^N C_{ik}^{m-1} v_i^m$. If $E^2 = 0$, the weight matrix remains unchanged.

2.2 The Dynamics

First, let us prove the following lemma concerning the dynamics of the autoassociative network described in the previous section.

Lemma 2.1. *As a result of free evolution, the autoassociative network with symmetric weights and zero threshold may converge to a cycle – dynamic attractor. In this case the dynamic attractor consists* <u>*exactly*</u> *of two states.*

Proof. Consider the "energy" function[2] defined as

$$E(t) \doteq -\frac{1}{2}\mathbf{Y}^T(t)[\mathbf{S}(t-1) - \mathbf{B}] = -\frac{1}{2}\sum_{i=1}^{N} |s_i(t-1) - b_i|. \tag{4}$$

Using Eq. 2, that $\mathbf{Y}^T\mathbf{S} = \mathbf{S}^T\mathbf{Y}$ and $\mathbf{C} = \mathbf{C}^T$ we have

$$\begin{aligned} \mathbf{Y}^T(t)\ [\mathbf{S}(t-1) - \mathbf{B}] &= \mathbf{Y}^T(t)\mathbf{C}\mathbf{Y}(t-1) - \mathbf{Y}^T(t)\mathbf{B} = \\ &\mathbf{Y}^T(t-1)\mathbf{S}(t) - \mathbf{Y}^T(t)\mathbf{B} = \mathbf{Y}^T(t+1)[\mathbf{S}(t) - \mathbf{B}] - \\ &[\mathbf{Y}^T(t+1) - \mathbf{Y}^T(t-1)] * [\mathbf{S}(t) - \mathbf{B}] - [\mathbf{Y}^T(t) - \mathbf{Y}^T(t-1)]\mathbf{B}. \end{aligned} \tag{5}$$

Substituting this equation into Eq. 4 yields:

$$E(t) = E(t+1) + \sum_{h=1}^{H(t-1,t+1)} |s_h(t) - b_h| + \sum_{h=1}^{H(t-1,t)} y_h(t) b_h, \tag{6}$$

where $H(t-1, t+1)$ is the number of such neurons y_h that $y_h(t-1) \neq y_h(t+1)$.

As can be seen, in the absence of the threshold (i.e. when $b_i = 0$) the "energy" function is monotonically decreasing,

$$E(t+1) < E(t) \tag{7}$$

And since there is a lower bound:

$$\inf(E(t)) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} |C_{ij}|, \tag{8}$$

in the long run the network will converge to a state such that the second term of Eq. 6 becomes zero. This is achieved when $\mathbf{Y}(t-1) = \mathbf{Y}(t+1)$. However, this does not imply that the subsequent state $\mathbf{Y}(t)$ is the same as $\mathbf{Y}(t-1)$.

Thus, out of all 2^N states, in which the network can be, there are some states $\mathbf{Y}^{st}$ which are static attractors, i.e. $\mathbf{Y}(t-2) \to \mathbf{Y}(t-1) = \mathbf{Y}(t) = \mathbf{Y}^{st}$, and there are also some states $\mathbf{Y}^{d1}$, $\mathbf{Y}^{d2}$ which form a dynamic attractor, i.e. $\mathbf{Y}(t-2) \to \mathbf{Y}^{d1} \to \mathbf{Y}^{d2} \to \mathbf{Y}^{d1} \to \mathbf{Y}^{d2}$. That is, an attractor of the autoassociative network is either a static attractor consisting of one stable state or a dynamic attractor consisting of two states. □

[2] Note that this "energy" function is different from the conventional energy [3, 6] defined as $E^{conv}(t) \doteq -\frac{1}{2}\mathbf{Y}^T(t)[\mathbf{S}(t) - \mathbf{B}]$.

Generally speaking, the fact that the cycles of autoassociative networks are of length 2 has been shown before by Frumkin and Moses [15], but their proof is based on the presumption of the deterministic nature of the evolution process. Also, it does not show what influences the occurrence of cycles. On the other hand, our proof of the Lemma shows this. In particular, one can see how the threshold influences the dynamics.

In the case of a non-zero threshold, due to the third term in Eq. 6 one can obtain that 1) if $b_h < 0.5|s_h|$, then the network converges to states $\mathbf{Y}$ which have components y_i such that $y_i b_i < 0$; and 2) if $b_h \geq 0.5|s_h|$, then the network does not converge at all. This allows the increase of the attraction radii of some attractors, which is the basis for the adjustable threshold networks (their description can be found in [16]).

2.3 Update flow technique

The probability of being trapped by a spurious attractor, either dynamic or static, increases with the number of prototypes [12, 14]. While the number of prototypes is small (much smaller than the capacity of the network), dynamic attractors may not be observed, as in [6, 7]. Yet, when the number of prototypes is large, their existence is observed more often (see [12]). In this case, in order to prevent infinite iterations, one needs to know the way of identifying them. The above lemma shows an easy way of doing it. All we need to do is to store the indices of the updated neurons till the next update. If in the next update they are the same, we are trapped in a dynamic attractor.

Previously, we introduced an *update flow technique* for updating a state of a network [17]. It is based on storing a buffer of neurons updated in a current iteration rather than a vector of all neurons. This technique is proved to be more preferable on account of its high evaluation speed and suitability for parallel implementation, which is due to processing only the data that has been changed since the last iteration. As we can see now, this technique has one more advantage — it does not need extra processing required for the above suggested checking for dynamic attractors.

3 Pseudo-Inverse Network With Reduced Self-Connection

3.1 Occurrence of Dynamic Attractors

In [4] it has been shown that the *pseudo-inverse network*, i.e. the network designed with the pseudo-inverse learning rule (Eq. 3), does not have cycles, despite the fact that its weights are symmetric[3]. Recently in [12, 13], it has been shown that partial reduction of self-connection described as

$$C'_{ii} = D \cdot C_{ii}, \quad (0 < D < 1); \quad C'_{ji} = C_{ji} \quad (i \neq j), \tag{9}$$

[3] In this section we consider the threshold equal to zero.

improves significantly the retrieval capability of this network. In particular, it increases attraction radii and allows the network to retrieve up to $0.75N$ prototypes.

However, the deviation from Eq. 3, results in changing the dynamics of the network — according to the lemma, dynamic attractors may now occur. And as our simulations show they do occur, especially for large M ($M > 0.6N$) and small D ($D \leq 0.1$). Let us show theoretically why this happens.

Theorem 3.1. *The number of dynamic attractors in the pseudo-inverse network increases with the number of prototypes M and the degree of weight reduction $\alpha \doteq 1 - D$.*

Proof. For $\{\mathbf{Y}^{d1}, \mathbf{Y}^{d2}\}$ to be a dynamic attractor the following conditions must be met

$$\begin{cases} y_i^{d1} \sum_{j=1}^{N} C'_{ij} y_j^{d1} < 0 \\ y_i^{d2} \sum_{j=1}^{N} C'_{ij} y_j^{d2} < 0 \end{cases} \text{for } \forall i \in \Omega, \tag{10}$$

where Ω is the set of the indices of oscillating neurons, i.e. $\Omega = \{i : y_i^{d1} = -y_i^{d2}\}$. Adding one equation to another we get

$$y_i^{d1}(\sum_{j=1}^{N} C'_{ij} y_j^{d1} - \sum_{j=1}^{N} C'_{ij} y_j^{d2}) = 2y_i^{d1} \sum_{j \in \{\Omega\}} C'_{ij} y_j^{d1} < 0 \tag{11}$$

Summing this equation over all oscillating neurons we have

$$2 \sum_{i \in \Omega} y_i^{d1} \sum_{j \in \Omega} C'_{ij} y_j^{d1} = 2 \sum_{i \in \Omega} \sum_{j \in \Omega} C'_{ij} y_j^{d1} y_i^{d1} < 0 \tag{12}$$

and using Eq. 9 we obtain

$$\sum_{i,j \in \Omega} C_{ij} y_j^{d1} y_i^{d1} < (1 - D) \sum_{i \in \Omega} C_{ii}. \tag{13}$$

Since for the pseudo-inverse learning rule $C_{ii} \sim \frac{M}{N}$ (see [12]) and the right-hand side of Eq. 13 is always positive, we obtain that the probability of $\{\mathbf{Y}^{d1}, \mathbf{Y}^{d2}\}$ to be a dynamic attractor is proportional to M and $\alpha \doteq 1 - D$, *q.e.d.* □

Whether or not the network will be trapped in a dynamic attractor mentioned by the Theorem, depends upon the number of states the network passes through during the evolution, i.e. the number of iterations. The further the initial state of the network is from its final state (i.e. the greater initial noise of a pattern to be retrieved), the more iterations it takes to reach this state. This explains why the number of cycle occurrences increases not only with M and α but also with the value of initial noise.

Experiments Figure 1 shows the probability of occurrence of dynamic attractors as a function of D and initial noise $H0$, as observed in simulations — in the area below the lines, they occur with a probability not greater than 1%. The simulations were carried out with a network of 100 neurons. M random vectors ($M = 20, 40, 60$), in which 40 out of 100 neurons were clipped randomly to be in the "+1" state and the rest of neurons were in the "−1" state, were used as prototypes. The initial state was obtained by randomly inverting $H0$ neurons of a prototype. The figure presents data averaged over 10 different prototype sets and 10 different noise implementations for each value of noise $H0$. As it can be seen, cycles are indeed observed more often for large values of M, small values of D and large values of noise $H0$.

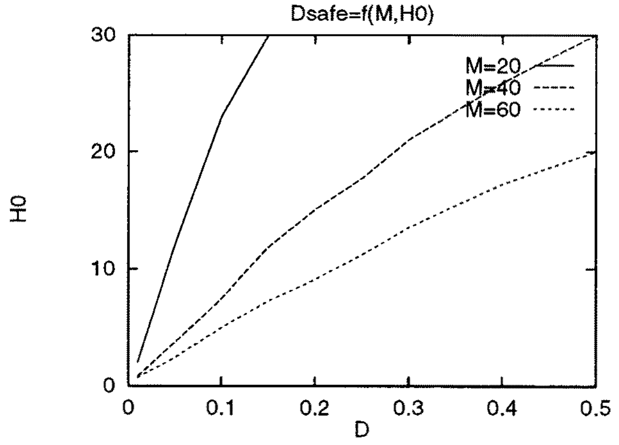

Figure 1. *The likelihood of occurrence of dynamic attractors as a function of D and initial noise H0. In the area below the lines they occur with a probability not greater than 1%.*

3.2 Predicting Dynamic Attractors

Simulations show that when a cycle occurs, the number of oscillating neurons in most cases equals two[4]. From Eq. 13 in the proof of the Theorem we can find for which D this happens. For the set Ω consisting of two indices i and j ($\Omega = \{i, j\}$) from Eq. 13 we have

$$2|C_{ij}| > D(C_{ii} + C_{jj}). \tag{14}$$

Using estimates of C_{ij} and C_{ii} obtained in [12]:

$$C_{ii} \sim \frac{M}{N}, \qquad C_{ij}^2 \sim \frac{M(N-M)}{N^3} \tag{15}$$

[4] To be more exact, the number of oscillating neurons can vary from one to almost N, although two neurons are observed in most cases except when $D = 0$.

we have for D: $D^2 < \frac{C_{ij}^2}{C_{ii}^2} = \frac{N-M}{NM}$. For $N = 100$ and $\frac{M}{N} = 0.5$, for example, we obtain that a cycle will most likely occur when $D < 0.1$, which is in agreement with the experimental observations. But occurrence of cycles also depends upon the initial noise, as was mentioned above.

In many applications the value of the initial noise, i.e. the distance from the initial state $\mathbf{Y}$ to the main attractors $\mathbf{V}^1, ..., \mathbf{V}^M$, is not known *a-priori*. Then in order to predict a cycle, we may use the Euclidian distance E between the vector $\mathbf{Y}$ and the subspace spanned by prototype vectors, which can be found (see [4]) as

$$E^2 \doteq ||\mathbf{CY} - \mathbf{Y}||^2 = N - \mathbf{Y}^T\mathbf{S} = N - \sum_{i=1}^{N} y_i s_i. \tag{16}$$

The value of E^2 ranges from 0 to N and its calculation does not require a lot of processing time. If E^2 appears to be large, meaning that the initial noise is large, than additional measures to handle cycles, like that mentioned in Section 2, can be undertaken.

Other methods to prevent cycles may include 1) using asynchronous dynamics which, although being slow, does not produce cycles [3, 10] or 2) using the parallel dynamics with memory terms suggested in [18], where instead of Eq. 1 the following synchronous update rule is used:

$$y_i(t+1) = \text{sgn}[\frac{1}{2}s_i(t) + \frac{1}{2}s_i(t-1)]. \tag{17}$$

For this rule, according to [18], cycles are very rare.

4 Conclusions

In this paper we have considered autoassociative neural networks, which are intensively used for pattern recognition and low-level computer vision problems, and studied the occurrence of cycles, which is an important problem concerning these networks. These cycles are referred to as dynamic attractors. We have shown theoretically when and why dynamic attractors occur. In particular, we proved a lemma that if a dynamic attractor occurs, then it consists of exactly two states. This led us to a technique which allows us to identify them.

We have also shown that there exist dynamic attractors in the pseudo-inverse network with reduced self-connection and that they occur with a probability proportional to the number of prototypes M and the degree of weight reduction $\alpha = 1 - D$. We demonstrated how to estimate a coefficient of self-connection reduction D which can lead to the occurrence of cycles and suggested an additional criteria, based on the distance between an initial state and the prototypes, which can be used to predict the occurrence of cycles.

References

1. S. Amari. Neural theory of association and concept formation, *Biological Cybernetics*, vol 26, pp. 175-185, 1977.

2. T. Kohonen. *Self-organization and associative memory*, Berlin:Springler, 1984.
3. J.J. Hopfield. Neural networks and physical systems with emergent collective computational abilities, *Proc. Nat. Acad. Sci. USA*, vol 79, pp. 2554-2558, 1982.
4. L. Personnaz, I. Guyon and G. Dreyfus. Collective computational properties of neural networks: New learning mechanisms, *Phys. Rev. A* vol 34, pp. 4217-4228, 1986.
5. M.A. Arbib and A.R. Hanson. *Vision, Brain, and Cooperative Computation*, The MIT Press, 1987.
6. Y. Zhou and R. Chellappa. *Artificial Neural Networks for Computer Vision*, Research Notes in Neural Computing, vol 5, Springer-Verlag, New-York, 1992.
7. D. Valentin, H. Abdi, A.J. O'Toole, G.W. Cottrell. Connectionist models of face processing: A survey. *Pattern Recognition*, vol 27, 1208-1230, 1994.
8. V.A. Morozov. *Methods for Solving Incorrectly Posed Problems*, Springer-Verlag, New-York, 1984.
9. D.O. Gorodnichy, W.W. Armstrong, X. Li. Adaptive Logic Networks for Facial Feature Detection, in these proceedings of *ICIAP'97*.
10. K. Cheung, L. Atlas, R. Marks. Synchronous vs asynchronous behavior of Hopfield's CAM neural net, *Applied Optics*, Vol. 26, No 22, pp. 4808-4813, 1987.
11. E. Goles, F. Fogelman, D. Pellegrin. Decreasing Energy Functions as a Tool for Studying Threshold Networks. *Discrete Appl. Math.*, no 12, pp. 261-277, 1985.
12. D.O. Gorodnichy. Desaturating Coefficient for Projection Learning Rule, *Intern. Conf. on Artificial Neural Networks (ICANN'96) Proceedings*, pp. 469-476, Bochum, Germany, Springer-Verlag, Lecture Notes in Computer Science 1112, 1996.
13. D.O. Gorodnichy and A.M. Reznik. Increasing Attraction of Pseudo-Inverse Autoassociative Networks, *Neural Processing Letters*, volume 5, issue 2, pp. 123-127, 1997.
14. A.M. Reznik. *Iterativnyǐ proekcionnyǐ algoritm obucheniya neǐronnykh seteǐ*, Kibernetika i sistemnyǐ analiz, in russian, (Резник А.М. Итеративный проекционный алгоритм обучения нейронных сетей, Кибернетика и системный анализ), no 6, pp. 131-141, 1993.
15. A. Frumkin, E. Moses. Physicality of the Little model, *Phys. Rev. A* vol 34, No 1, pp. 714-716, July 1986.
16. A. Schultz. Five variations of Hopfield Associative memory networks, *Journal of Artificial Neural Networks* vol. 2, no 3, pp. 285-294, 1995.
17. D.O. Gorodnichy, A.M. Reznik. NEUTRAM — A Transputer Based Neural Network Simulator, *Proc. of Second Intern. Conf. on Software for Multiprocessors and Supercomputers Theory, Practice, Experience (SMS TPE'94)*, pp. 136-142, Moscow, Russia, 1994.
18. I. Kanter and H. Sampolinsky. Associative recall of memory without errors, *Phys. Rev. A* vol 35, pp. 380-392, 1987.