INTERNATIONAL COMPUTER SCIENCE INSTITUTE

1947 Center Street • Suite 600 • Berkeley, California 94704 • 1-510-642-4274 • FAX 1-510-643-7684

On Some Stability Properties of the LRAAM Model

Alessandro Sperduti

TR-93-031

July 1993

Abstract

In this report we discuss some mathematical properties of the LRAAM model. The LRAAM model is an extension of the RAAM model by Pollack. It allows one to obtain distributed reduced representations of labeled graphs. In particular, we give sufficient conditions on the asymptotical stability of the decoding process along a cycle of the encoded structure.

Data encoded in an LRAAM can also be accessed by content by transforming the LRAAM in an analog Hopfield network with hidden units and asymmetric connection matrix (**CA** network.) Different access procedures can be defined according to the access key. Each access procedure corresponds to a particular constrained version of the **CA** network. We give sufficient conditions under which the property of asymptotical stability of a fixed point in one particular constrained version of the **CA** network can be extended to related fixed points of different constrained versions of the **CA** network. An example of encoding of a labeled graph on which the theoretical results are applied is given as well.

1 Introduction

The concept of distributed reduced representations was introduced by Hinton [4] in order to allow a neural network to represent compositional structure (see also [12, 15, 17]). Concrete examples of distributed reduced representations are given by the Recursive Auto-Associative Memory (RAAM) by Pollack [11, 12] and by the Holographic Reduced Representations of Plate [10]. In particular, the RAAM model is able to generate reduced representations of lists and fixed-valence trees with information stored in the leaves. The Labeling RAAM model (LRAAM) [16] has extended the RAAM model, allowing the synthesis of distributed reduced representations for fixed-valence labeled graphs. Another advantage of the LRAAM model is that it allows one to retrieve information both by using the reduced representations and by content. The last capability is obtained by exploiting the structure of the LRAAM network which can be easily transformed in an analog Hopfield network [5] with hidden units. Because of the structure of each pattern in the training set, several constrained versions of the modified LRAAM network can be used, according to the access key, in order to improve the retrieval by content. In recent years, several results related to the analog Hopfield model have been published, both without considering hidden units [3, 8, 2, 14] and with hidden units [9, 7, 1].

In this report, we discuss conditions for the stability of equilibria arising in the LRAAM model. Two different kinds of stability problems are present: when considering the decoding of a pointer along a cycle of the encoded structures and when considering the stability of the equilibria in the original and constrained versions of the modified network used for the retrieval of data by content. The main contribution of this report is to give sufficient conditions for the asymptotical stability of such model. Moreover, the report shows how the confluence of distributed reduced representations and analog Hopfield networks in the LRAAM model leads to an advance in both fields: theoretical analysis of networks involving distributed reduced representations and a learning procedure to store complex structures in a Hopfield network, besides to new specialized theoretical results, are now available.

The report is organized as follows. In Section 2 we introduce informally the LRAAM model. A formal setting for the model and the stability problems discussed above are presented in Section 3. The decoding problem and the stability problems for the retrieval of data by content are faced respectively in Section 4 and Section 5. An example of encoding of a labeled graph is given in Section 6, where the theorems presented in the previous sections are applied and experimentally verified. Conclusions are drawn in Section 7.

2 The LRAAM model

The *Labeling RAAM (LRAAM)* is an extension of the RAAM model which allows one to encode labeled structures. The general structure of the network for an LRAAM is shown in Figure 1.

The network is trained by backpropagation [13] to learn the identity function. The idea is to obtain a compressed representation (hidden layer activation) of a node of a labeled graph by allocating a part of the input (output) of the network to represent the label (N_l units) and the rest to represent one or more pointers. The compressed representation is then used as pointer to the node. In order to allows the recursive use of such compressed representations, the part of the input (output) layer which represents a pointer must be of the same dimension as the hidden layer (N_H units). Thus, a general LRAAM is implemented by a $N_I - N_H - N_I$ feed-forward network, where

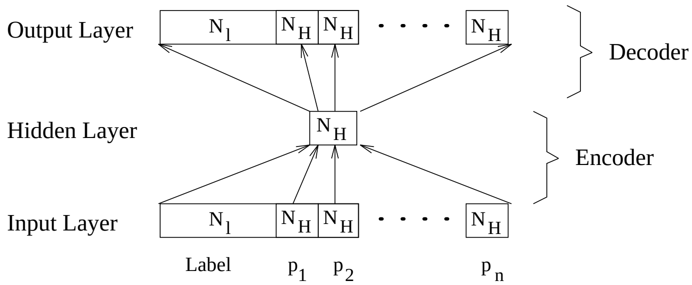

Figure 1: The network for a general LRAAM. The first layer of the network implements an encoder; the second layer, the corresponding decoder.

$N_I = N_l + nN_H$, and n is the number of pointer fields.

Labeled graphs can be easily encoded using an LRAAM. It suffices to represent each node in the graph as a record with one field for the label and one field for each pointer to a connected node. The pointers need to be only logical pointers, since their actual values will be the patterns of hidden activation of the network. At the beginning of learning, their values are set at random. A graph is represented by a list of such records, and such a list constitutes the training set for the LRAAM.

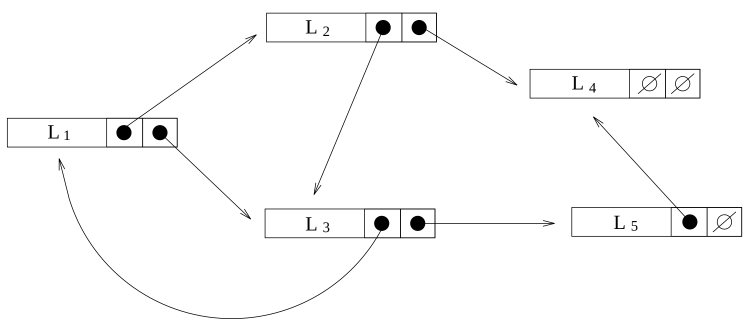

Figure 2: Example of labeled graph.

For example, the network for the graph shown in Figure 2 can be trained as follows:

input		hidden		output
$(L_1\ P_{n2}(t)\ P_{n3}(t))$	$\rightarrow$	$P_{n1}(t)$	$\rightarrow$	$(L_1'(t)\ P_{n2}'(t)\ P_{n3}'(t))$
$(L_2\ P_{n3}(t)\ P_{n4}(t))$	$\rightarrow$	$P_{n2}(t)$	$\rightarrow$	$(L_2'(t)\ P_{n3}'(t)\ P_{n4}'(t))$
$(L_3\ P_{n1}(t)\ P_{n5}(t))$	$\rightarrow$	$P_{n3}(t)$	$\rightarrow$	$(L_3'(t)\ P_{n1}'(t)\ P_{n5}'(t))$
$(L_4\ nil\ nil)$	$\rightarrow$	$P_{n4}(t)$	$\rightarrow$	$(L_4'(t)\ nil'(t)\ nil'(t))$
$(L_5\ P_{n4}(t)\ nil)$	$\rightarrow$	$P_{n5}(t)$	$\rightarrow$	$(L_5'(t)\ P_{n4}'(t)\ nil'(t))$

where L_i and P_{ni} are respectively the label of and the pointer to the ith node and t represents the time, or epoch, of training. If the backpropagation algorithm converges to perfect learning, i.e., the total error goes to zero, it can be stated that:

$$
\begin{array}{ccc}
L_1 = L_1' & L_2 = L_2' & L_3 = L_3' \\
L_4 = L_4' & L_5 = L_5' & P_{n2} = P_{n2}' \\
P_{n3} = P_{n3}' & P_{n4} = P_{n4}' & P_{n5} = P_{n5}'
\end{array}
$$

Once the training is complete, the patterns of activation representing pointers can be used to retrieve information. Thus, for example, if the activity of the hidden units of the network is clamped to P_{n1}, the output of the network becomes (L_1,P_{n2},P_{n3}), permitting further retrieval of information by decoding P_{n2} or P_{n3}, and so on. Note that more labeled graphs can be encoded in the same LRAAM.

One advantage of the LRAAM model, besides to return *distributed reduced representations* of labeled graphs, is that information can be accessed by content as well. In fact, if the output of the network is fed back to the input, the network is transformed in an analog Hopfield network with one hidden layer and asymmetric connection matrix. Now, because the network was trained to implement the identity function, it turns out that each pattern of the training set[1] is a fixed point of the transformed network, provided that the residual error is below a training set dependent tolerance. In general, not all the patterns are asymptotically stable fixed points. However, since we are dealing with structured patterns, it is possible to exploit such a structure during the relaxation of the network in order to improve the retrieval of data. In fact, a fixed point for the unconstrained network which is not asymptotically stable may become stable in a constrained version of the network[2]. In particular, because each pattern is structured in different fields, different access procedures can be defined on the Hopfield network according to the type of access key. An access procedure is defined by:

1. choosing one or more fields in the input layer according to the access key(s);

2. clamping the output of such units to the known information, i.e., the access key(s);

3. randomly setting the output of the remaining units in the network;

4. letting the unclamped units of the network to relax into a stable state.

A validation test of the reached stable state can be performed by:

1. unfreezing the clamped units in the input layer;

2. if the stable state is no longer stable the result of the procedure is considered wrong and another run is performed;

3. otherwise the stable state is considered a success.

This validation test, however, can sometimes fail to detect an erroneous retrieval because of the existence of spurious stable states of the constrained network that share the same known information with the desired one.

In the next section we give a formal description of both the feed-forward network (Pointer Access network) and the recurrent network (Content Access network) defined for an LRAAM. Related asymptotical stability problems are introduced.

[1] To be precise, the final version of the training set, since it is dynamical.

[2] We will see that the converse is also true, i.e., an asymptotically stable fixed point of the unconstrained network may became unstable for the constrained version.

3 Formal description

The Pointer Access (PA) network:

$$\boldsymbol{F}_E(\vec{x}) = \boldsymbol{F}(\boldsymbol{E}\vec{x} + \vec{\theta}_H) \equiv \vec{h}, \tag{1}$$

$$\boldsymbol{F}_D(\vec{h}) = \boldsymbol{F}(\boldsymbol{D}\vec{h} + \vec{\theta}_O) \equiv \vec{o}, \tag{2}$$

where $\vec{h}, \vec{\theta}_H \in \Re^{N_H}$, $\vec{o}, \vec{x}, \vec{\theta}_O \in \Re^{N_H+nN_H}$, $\boldsymbol{F}_i(\vec{x}) = f(x_i)$, and $f()$ is a sigmoid-shaped function;

The Content Access (CA) network:

$$\textbf{PA}\ \textit{network}\ +\ \vec{o} \equiv \vec{x}. \tag{3}$$

In the **PA** network, $\boldsymbol{E} \in \Re^{N_H \times N_I}$ is the Encoding matrix, i.e., the weight matrix between the input layer ($N_I = N_l + nN_H$ units) and the hidden layer (N_H units), and $\boldsymbol{D} \in \Re^{N_O \times N_H}$ is the Decoding matrix, i.e., the weight matrix between the hidden layer and the output layer ($N_O = N_I$ units). In order to make explicit the partition of the input and output layers, $\boldsymbol{E}$ and $\boldsymbol{D}$ can be written as the composition of submatrices, one for each field of the LRAAM:

$$\boldsymbol{E} = [\boldsymbol{E}^{(l)}, \boldsymbol{E}^{(p_1)}, \ldots, \boldsymbol{E}^{(p_n)}], \tag{4}$$

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{D}^{(l)} \\ \boldsymbol{D}^{(p_1)} \\ \vdots \\ \boldsymbol{D}^{(p_n)} \end{bmatrix}, \tag{5}$$

where $\boldsymbol{E}^{(l)} \in \Re^{N_H \times N_l}$ is the weight matrix between the label field of the input layer and the hidden layer, $\boldsymbol{D}^{(l)} \in \Re^{N_l \times N_H}$ is the weight matrix between the hidden layer and the label field of the output layer, $\boldsymbol{E}^{(p_1)} \in \Re^{N_H \times N_H}$ is the weight matrix between the first pointer field of the input layer and the hidden layer, and so on. Following this notation, also the bias, input, and output vectors can be partitioned in subvectors:

$$\vec{\theta}_H = [\vec{\theta}_H^{(l)^t}, \vec{\theta}_H^{(p_1)^t}, \ldots, \vec{\theta}_H^{(p_n)^t}]^t, \tag{6}$$

$$\vec{\theta}_O = [\vec{\theta}_O^{(l)^t}, \vec{\theta}_O^{(p_1)^t}, \ldots, \vec{\theta}_O^{(p_n)^t}]^t, \tag{7}$$

$$\vec{x} = [\vec{x}^{(l)^t}, \vec{x}^{(p_1)^t}, \ldots, \vec{x}^{(p_n)^t}]^t, \tag{8}$$

$$\vec{o} = [\vec{o}^{(l)^t}, \vec{o}^{(p_1)^t}, \ldots, \vec{o}^{(p_n)^t}]^t, \tag{9}$$

where t is the transpose operator. Moreover, due to the constraints over the **PA** network, the dimension of each subvector referring to a pointer field must be equal to the dimension of $\vec{h}$.

The **CA** network is obtained by the **PA** network simply by connecting the output to the input ($\vec{o} \equiv \vec{x}$), and it actually constitutes a discrete time Hopfield network with analog output and asymmetric connection matrix. The underlying idea is that each pattern of the training set, assuming zero total error after learning, becames a fixed point of the **CA** network. Moreover, under some conditions these fixed points are also asymptotically stable. Thus, if only partial information is known about a pattern, the **CA** network can be used to reconstruct the whole pattern. Since each pattern is structured, it turns out that it is best to take advantage of this structure by constraining the relaxation process of the **CA** network. The constraints are obtained, as informally discussed above, by considering only subsets of connections of the **CA** network, according to the known information.

In this report we focus on two stability problems encountered in the LRAAM model. The first one arises when considering the decoding of a pointer along a cycle of the encoded structures. Since the decoding process suffers, in general, of approximation errors, it can happen that the decoding may diverge from the correct representations of the pointers belonging to the cycle. Thus, it is fundamental to discover under which conditions the representations obtained for the pointers belonging to a cycle are asymptotically stable with respect to the pointer transformation. In fact, if the representations are asymptotically stable, the errors introduced by the decoding function are automatically corrected.

The second problem consists of discovering sufficient conditions under which the property of asymptotical stability of a fixed point in one particular constrained version of the **CA** network could be extended to related fixed points of different constrained versions of the **CA** network.

4 The Decoding Problem

Given a pointer to a component of the encoded structure, it is possible to access the information contained in it by using the Decoding function $\boldsymbol{F}_D()$ (eqn. 2). It is convenient, for our purposes, to make explicit the single decoding transformation implemented by each submatrix of $\boldsymbol{D}$. The pointer transformations are defined as follows:

$$\forall j,\ j = 1, \ldots, n,\ \boldsymbol{F}^{(p_j)}(\vec{d}\,) = \boldsymbol{F}(\boldsymbol{D}^{(p_j)}\vec{d} + \vec{\theta}_O^{(p_j)}), \tag{10}$$

and the label decoding function as:

$$\boldsymbol{F}^{(l)}(\vec{d}\,) = \boldsymbol{F}(\boldsymbol{D}^{(l)}\vec{d} + \vec{\theta}_O^{(l)}). \tag{11}$$

Let $\boldsymbol{J}^{(p_j)}(\vec{d}\,) = \boldsymbol{\Lambda}(\vec{net}^{(p_j)})\boldsymbol{D}^{(p_j)}$ be the Jacobian matrix of the transformation $\boldsymbol{F}^{(p_j)}(\vec{d}\,)$, where $\vec{net}^{(p_j)} = \boldsymbol{D}^{(p_j)}\vec{d} + \boldsymbol{\theta}_O^{(p_j)}$ and $\boldsymbol{\Lambda}(\vec{net}^{(p_j)}) \in \Re^{N_H \times N_H}$ is the diagonal matrix whose ith diagonal element is $f'(\vec{net}_i^{(p_j)}) \equiv f_i'(\vec{d})$. Then the following theorem holds:

Theorem 1 *A decoding sequence*

$$\vec{d}^{\,(i_{j+1})} = \boldsymbol{F}^{(p_{i_j})}(\vec{d}^{\,(i_j)}), \quad j = 0, \ldots, L \tag{12}$$

with $\vec{d}^{\,(i_{L+1})} = \vec{d}^{\,(i_0)}$, *satisfying*

$$\sum_{k=1}^{N_H} |b_{ik}| < 1, \qquad i = 1, \ldots, N_H \tag{13}$$

for some index p_{i_q}, $q = 0, \ldots, L$, *is asymptotically stable, where* b_{ik} *is the* (i,k)*th element of a matrix* $\boldsymbol{B}$, *given by*

$$\boldsymbol{B} = \boldsymbol{J}^{(p_{i_q})}(\vec{d}^{\,(i_q)})\boldsymbol{J}^{(p_{i_{q-1}})}(\vec{d}^{\,(i_{q-1})}) \cdots \boldsymbol{J}^{(p_{i_0})}(\vec{d}^{\,(i_0)})\boldsymbol{J}^{(p_{i_L})}(\vec{d}^{\,(i_L)}) \cdots \boldsymbol{J}^{(p_{i_{q+1}})}(\vec{d}^{\,(i_{q+1})}).$$

Proof: By definition, $\vec{d}^{\,(i_0)}$ is a fixed point of the transformation:

$$\boldsymbol{G}^{(p_{i_0})}(\vec{d}\,) = \boldsymbol{F}^{(p_{i_L})}(\boldsymbol{F}^{(p_{i_{L-1}})}(\cdots \boldsymbol{F}^{(p_{i_0})}(\vec{d}\,)\cdots)), \tag{14}$$

and the sequence in eqn.12 is asymptotically stable if and only if $\vec{d}^{\,(i_0)}$ is an asymptotically stable fixed point of eqn.14. The linearized system for eqn.14 at $\vec{d}^{\,(i_0)}$ is:

$$\vec{x}^{\,(t+1)} = \boldsymbol{B}(\vec{d}^{\,(i_0)})\vec{x}^{\,(t)}, \tag{15}$$

where $\vec{x}^{\,(t)} = \vec{d}^{\,(t)} - \vec{d}^{\,(i_0)}$, and $\boldsymbol{B}(\vec{d}^{\,(i_0)})$ is the Jacobian matrix of $\boldsymbol{G}^{(p_{i_0})}(\vec{d}^{\,(i_0)})$. On differentiating a function of functions, it turns out that:

$$\boldsymbol{B}(\vec{d}^{\,(i_0)}) = \boldsymbol{J}^{(p_{i_L})}(\vec{d}^{\,(i_L)})\cdots \boldsymbol{J}^{(p_{i_0})}(\vec{d}^{\,(i_0)}). \tag{16}$$

Let λ be an eigenvalue of $\boldsymbol{B}(\vec{d}^{\,(i_0)})$, associated with eigenvector $\vec{v}$. Then it is easy to verify that λ is also an eigenvalue of $\boldsymbol{B}(\vec{d}^{\,(i_j)})$, associated with eigenvector $\boldsymbol{J}^{(p_{i_j})}(\vec{d}^{\,(i_j)})\cdots \boldsymbol{J}^{(p_{i_0})}(\vec{d}^{\,(i_0)})\vec{v}$. By the fact that $\vec{d}^{\,(i_0)}$ is asymptotically stable if and only if all the eigenvalues associated to $\boldsymbol{G}^{(p_{i_0})}(\vec{d}^{\,(i_0)})$ have moduli less than one, and by using the Gershgorin's Theorem, the assertion of the theorem is demonstrated. □

Actually, this theorem is a discrete time adaptation of the Theorem 2 presented in [1] by Atiya and Abu-Mostafa, where the pointer transformations can be assimilated to different hidden layers of an appropriate network. Note that if at least one pointer, say $\vec{d}^{\,(x)}$, belonging to a cycle has saturated components, then the cycle is guaranteed to be stable. This is a direct consequence of the fact that the diagonal matrix $\boldsymbol{\Lambda}(\vec{net}^{(x)})$, which compounds the corresponding Jacobian matrix, is null or very close to null.

5 Retrieval of Data

The theorem stated in the previous section can be applied to guarantee the asymptotical stability of a constrained fixed point of the LRAAM, i.e., the fixed point obtained by clamping the output of a subset of the units to the key used to retrieve information. In order to discuss the issue formally, we need a bit of extra notation. Let us rename the components of $\boldsymbol{E}$ and $\boldsymbol{D}$ in the following way:

$$[\boldsymbol{E}^{(l)}, \boldsymbol{E}^{(p_1)}, \ldots, \boldsymbol{E}^{(p_n)}] = [\boldsymbol{E}^{(2^0)}, \boldsymbol{E}^{(2^1)}, \ldots, \boldsymbol{E}^{(2^n)}], \tag{17}$$

$$\begin{bmatrix} \boldsymbol{D}^{(l)} \\ \boldsymbol{D}^{(p_1)} \\ \vdots \\ \boldsymbol{D}^{(p_n)} \end{bmatrix} = \begin{bmatrix} \boldsymbol{D}^{(2^0)} \\ \boldsymbol{D}^{(2^1)} \\ \vdots \\ \boldsymbol{D}^{(2^n)} \end{bmatrix}. \tag{18}$$

Then, we can define a family of matrices given by subsets of the components of $\boldsymbol{E}$:

$$\boldsymbol{E}^{(x)} = [\boldsymbol{E}^{(\bar{x}_0)}, \ldots, \boldsymbol{E}^{(\bar{x}_{(k-1)})}], \tag{19}$$

where $\bar{x}_i = 2^j$ and b_j is the $(i+1)-th$ nonzero bit in the binary representation of x. Similarly, we can define the family of complementary matrices:

$$\hat{\boldsymbol{E}}^{(x)} = [\boldsymbol{E}^{(\hat{x}_0)}, \ldots, \boldsymbol{E}^{(\hat{x}_{(n-k)})}], \tag{20}$$

where $\hat{x}_i = 2^j$ and b_j is the $(i+1)-th$ zero bit in the binary representation of x. Note that, in general, $\boldsymbol{E}^{(x)} = \hat{\boldsymbol{E}}^{(y)}$ where y is the number obtained by complementing the binary representation of x. In particular, $\boldsymbol{E}^{(2^{n+1}-1)} = \hat{\boldsymbol{E}}^{(0)} = \boldsymbol{E}$ and $\boldsymbol{E}^{(0)} = \hat{\boldsymbol{E}}^{(2^{n+1}-1)} = \emptyset$. Thus, for example, if $\boldsymbol{E} = [\boldsymbol{E}^{(l)}, \boldsymbol{E}^{(p_1)}, \boldsymbol{E}^{(p_2)}]$, we have that:

$$\boldsymbol{E}^{(1)} = \boldsymbol{E}^{(l)}, \tag{21}$$

$$\hat{\boldsymbol{E}}^{(1)} = [\boldsymbol{E}^{(p_1)}, \boldsymbol{E}^{(p_2)}], \tag{22}$$

$$\boldsymbol{E}^{(2)} = \boldsymbol{E}^{(p_1)}, \tag{23}$$

$$\hat{\boldsymbol{E}}^{(2)} = [\boldsymbol{E}^{(l)}, \boldsymbol{E}^{(p_2)}], \tag{24}$$

$$\boldsymbol{E}^{(3)} = [\boldsymbol{E}^{(l)}, \boldsymbol{E}^{(p_1)}], \tag{25}$$

$$\hat{\boldsymbol{E}}^{(3)} = \boldsymbol{E}^{(p_2)}, \tag{26}$$

and so on.

The same notation can be used to derive a family of matrices by $\boldsymbol{D}$ and to represent a family of vectors composed of at most $n+1$ subvectors:

$$\vec{k} = [\vec{k}^{(l)^t}, \vec{k}^{(p_1)^t}, \ldots, \vec{k}^{(p_n)^t}]^t = [\vec{k}^{(2^0)^t}, \vec{k}^{(2^1)^t}, \ldots, \vec{k}^{(2^n)^t}]^t, \tag{27}$$

$$\vec{k}^{(x)} = [\vec{k}^{(\bar{x}_0)^t}, \ldots, \vec{k}^{(\bar{x}_{(k-1)})^t}]^t, \tag{28}$$

$$\hat{\vec{k}}^{(x)} = [\vec{k}^{(\hat{x}_0)^t}, \ldots, \vec{k}^{(\hat{x}_{(k-1)})^t}]^t, \tag{29}$$

where t is the transpose operator. The components of each subvector can be interpreted as associated to the units of one field of the LRAAM.

Now we are ready to define the equations driving the network used to retrieve information by the key vector $\vec{k}^{(x)}$:

$$\boldsymbol{F}_{\hat{E}^{(x)}}(\vec{\hat{o}}^{(x)}(t)) = \boldsymbol{F}(\hat{\boldsymbol{E}}^{(x)} \vec{\hat{o}}^{(x)}(t) + \vec{\theta}_H + \boldsymbol{E}^{(x)} \vec{k}^{(x)}) \equiv \vec{h}(t), \tag{30}$$

$$\boldsymbol{F}_{\hat{D}^{(x)}}(\vec{h}(t)) = \boldsymbol{F}(\hat{\boldsymbol{D}}^{(x)} \vec{h}(t) + \vec{\hat{\theta}}_O^{(x)}) \equiv \vec{\hat{o}}^{(x)}(t+1). \tag{31}$$

Note that $\boldsymbol{E}^{(x)} \vec{k}^{(x)}$ can be considered as an external input to the network, however, since it is constant during the retrieval by key $\vec{k}^{(x)}$, we prefer to consider it as an additional bias term.

Given a fixed point for the above equations, Theorem 1 can be applied directly to the composition of the functions $\boldsymbol{F}_{\hat{E}^{(x)}}()$ and $\boldsymbol{F}_{\hat{D}^{(x)}}()$. The same theorem can also be used to demonstrate some interesting relationships among the different networks we can derive by the **CA** network using our notation. The networks we derive are generated according to two criteria. First of all, it is decided which fields must be kept fixed, then it is decided which value for the key must be used. The first choice corresponds to fix the structure of the network by selecting the matrices $\hat{\boldsymbol{E}}^{(x)}$ and $\hat{\boldsymbol{D}}^{(x)}$, the second one to fix the bias of the hidden layer $(\vec{\theta}_H + \boldsymbol{E}^{(x)} \vec{k}^{(x)})$. Thus we can describe a network

X by the couple $(x, \vec{k}^{\,(x)})$, where x defines the structure of the network according to the original one, and $\vec{k}^{\,(x)}$ is the actual key vector used to retrieve the information. We say that a network Y is a subnetwork of network X ($Y \subset X$) if Y has been generated by fixing at least the same fields fixed to generate X. Note that, we have several networks[3] with the same structure but different bias values at the hidden layer.

Therefore, it is interesting to study how the asymptotically stability of a fixed point changes by changing the structure of the network or how the stability properties of a fixed point in a network are related to the stability properties of another fixed point in a network with the same structure but different bias values at the hidden layer. In the following we give three theorems regarding these aspects.

The first theorem can be considered a preservation theorem:

Theorem 2 *Given a fixed point* $(\vec{d}_H, \vec{\hat{d}}_O^{(x)})$ *for the network* $X \equiv (x, \vec{k}^{\,(x)})$ *satisfying the asymptotical stability condition (13), then the fixed point* $(\vec{d}_H, \vec{\hat{d}}_O^{(y)})$*, defined for the network* $Y \equiv (y, [\vec{k}^{\,(x)^t}, \vec{z}^{\,(y-x)^t}]^t)$*, with* $Y \subset X$*, and* $\vec{z}^{\,(y-x)}$ *given by the components of* $\vec{\hat{d}}_O^{(x)}$ *not in* $\vec{\hat{d}}_O^{(y)}$*, is asymptotically stable if:*

the stability condition (13) for $(\vec{d}_H, \vec{\hat{d}}_O^{(x)})$ *is satisfied at layer* O*;*

or

$\hat{\boldsymbol{E}}^{(x)} = \hat{\boldsymbol{D}}^{(x)^t} \hat{\boldsymbol{A}}^{(x)}$, *where* $\hat{\boldsymbol{A}}^{(x)}$ *is every diagonal matrix with components all of the same sign.*

Proof: Suppose the stability condition for $(\vec{d}_H, \vec{\hat{d}}_O^{(x)})$ is satisfied at layer O. First of all note that the columns of the Jacobian matrix $\boldsymbol{J}^{(\hat{E}^{(y)},Y)}(\vec{\hat{d}}_O^{(y)})$ for the network Y are a subset of the set of columns of the Jacobian matrix $\boldsymbol{J}^{(\hat{E}^{(x)},X)}(\vec{\hat{d}}_O^{(x)})$ for the network X, and the rows of the Jacobian matrix $\boldsymbol{J}^{(\hat{D}^{(y)},Y)}(\vec{d}_H)$ are a subset of the set of rows of the Jacobian matrix $\boldsymbol{J}^{(\hat{D}^{(x)},X)}(\vec{d}_H)$. Thus, since the stability condition (13) for $(\vec{d}_H, \vec{\hat{d}}_O^{(x)})$ and network X is satisfied with $\boldsymbol{B}^{(O,X)} = \boldsymbol{J}^{(\hat{D}^{(x)},X)}(\vec{d}_H)\boldsymbol{J}^{(\hat{E}^{(x)},X)}(\vec{\hat{d}}_O^{(x)})$, it follows that condition (13) is satisfied as well for $(\vec{d}_H, \vec{\hat{d}}_O^{(y)})$ and network Y because the elements of a row of the matrix $\boldsymbol{B}^{(O,Y)} = \boldsymbol{J}^{(\hat{D}^{(y)},Y)}(\vec{d}_H)\boldsymbol{J}^{(\hat{E}^{(y)},Y)}(\vec{\hat{d}}_O^{(y)})$ are a subset of the set of elements of a row of $\boldsymbol{B}^{(O,X)}$ which by hypothesis satisfy condition (13).

If the stability condition for $(\vec{d}_H, \vec{\hat{d}}_O^{(x)})$ is satisfied at layer H, then there is no guarantee that the modulus of an element of the matrix $\boldsymbol{B}^{(H,Y)} = \boldsymbol{J}^{(\hat{E}^{(y)},Y)}(\vec{\hat{d}}_O^{(y)})\boldsymbol{J}^{(\hat{D}^{(y)},Y)}(\vec{d}_H)$ will be less or equal than the modulus of the corresponding element of $\boldsymbol{B}^{(H,X)} = \boldsymbol{J}^{(\hat{D}^{(x)},X)}(\vec{d}_H)\boldsymbol{J}^{(\hat{E}^{(x)},X)}(\vec{\hat{d}}_O^{(x)})$, unless $\hat{\boldsymbol{E}}^{(x)}$ is equal to $\hat{\boldsymbol{D}}^{(x)^t} \hat{\boldsymbol{A}}^{(x)}$, where $\hat{\boldsymbol{A}}^{(x)}$ is every diagonal matrix with components all of the same sign. In this case, it is guaranteed that each element of $\boldsymbol{B}^{(H,X)}$ is given by the sum of quantities with the same sign and thus removing some of them will lead to a reduction of the modulus of the corresponding element of $\boldsymbol{B}^{(H,Y)}$. □

[3] One for each defined combination of values of the fixed fields.

In order to show that in general the satisfaction of condition (13) at layer H can lead to unstable fixed points when considering the constrained networks, in Figure 3 we have reported a simple network with this very unpleasant property. The value of the bias for each unit is shown on its side and $f(x) = \frac{2}{1+e^{-x}} - 1$. The network is constructed in such a way that $(f(0.95), [0.5, 0.5])$ is an asymptotically stable fixed point, as can be verified by computing the condition (13) at layer H. However, when constraining the network on x ($s_1 = -7 + \frac{7.9}{2}$) or on y ($s_1 = -7 + \frac{8}{2}$), the corresponding fixed point $(f(0.95), 0.5)$ turns unstable.

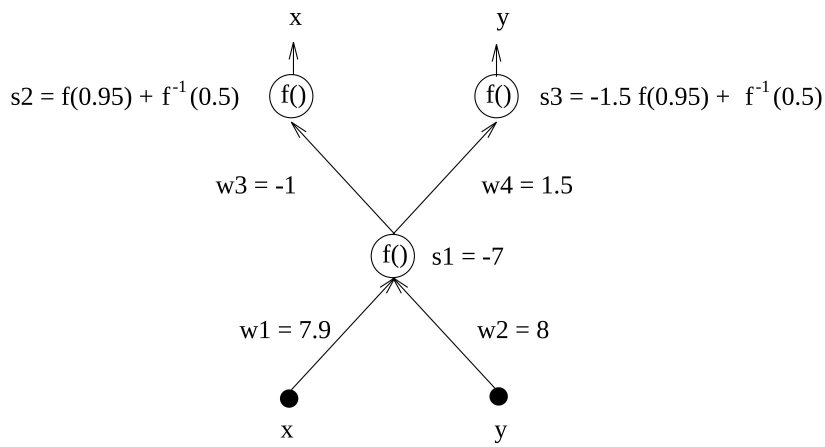

Figure 3: Example of network with an asymptotically stable fixed point but unstable related fixed points for the constrained networks.

The proof of the previous theorem gives also an idea of how an unstable fixed point of a network may be a stable fixed point of a subnetwork. In fact, it may happen that the elements of $\boldsymbol{B}$ are reduced in modulus or became null, allowing the stability condition to be satisfied. A particular case in which such event happens is captured in the following corollary.

Corollary 1 *Given a fixed point* $(\vec{d}_H, \vec{\hat{d}}_O^{(x)})$ *for the network* $X \equiv (x, \vec{k}^{(x)})$*, the fixed point* $(\vec{d}_H, \vec{\hat{d}}_O^{(y)})$*, defined for the network* $Y \equiv (y, [\vec{k}^{(x)^t}, \vec{z}^{(y-x)^t}]^t)$*, with* $Y \subset X$*, and* $\vec{z}^{(y-x)}$ *given by the components of* $\vec{\hat{d}}_O^{(x)}$ *not in* $\vec{\hat{d}}_O^{(y)}$*, is asymptotically stable if the stability condition (13) for* $(\vec{d}_H, \vec{\hat{d}}_O^{(x)})$ *is satisfied at layer* O *for all the indexes* i *which correspond to the components of* $\vec{\hat{d}}_O^{(y)}$*;*

Proof: By the first part of the proof of Theorem 2 all the rows of the matrix $\boldsymbol{B}^{(O,Y)} = \boldsymbol{J}^{(\hat{D}^{(y)},Y)}(\vec{d}_H)\boldsymbol{J}^{(\hat{E}^{(y)},Y)}(\vec{\hat{d}}_O^{(y)})$ satisfy condition (13) by hypothesis. □

The next two theorems regard the relationship between two fixed points of networks with the same structure but different bias vector at the hidden layer.

Theorem 3 *Given a fixed point* $(\vec{d}_H, \vec{\hat{d}}_O^{(x)})$ *for the network* $X \equiv (x, \vec{k}^{(x)})$ *satisfying the asymptotical stability condition (13), then the fixed point* $(\vec{d'}_H, \vec{\hat{d}}_O^{(x)})$*, defined for the network* $X' \equiv (x, \vec{k'}^{(x)})$*, is asymptotically stable if:*

the stability condition (13) for $(\vec{d}_H, \vec{\hat{d}}_O^{(x)})$ *is satisfied at layer* H*,*

and

$\forall i,\ i = 1, \ldots, N_H,\ f_i'(\vec{d'}_H) \ \leq \ f_i'(\vec{d}_H)$.

Proof: The assertion is a direct consequence of the fact that only the diagonal matrix referring to the first derivatives of the outputs of the hidden layer changes, and the change is such to reduce the moduli of the elements of $\boldsymbol{B}^{(H,X')}$ when considering the fixed point $(\vec{d'}_H, \vec{\hat{d}}_O^{(x)})$ since $f'()$ is positive definite. □

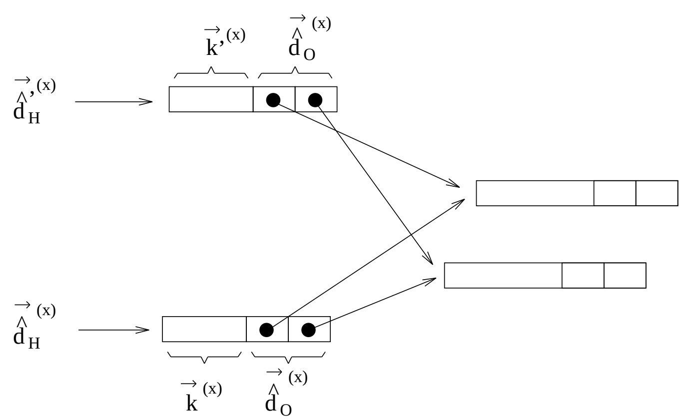

Figure 4: Example of application of Theorem 3.

The previous theorem is useful, for example, when we have more nodes with the same pointers but different labels (see Figure 4). In this case, if one among these nodes exists which is asymptotically stable and whose pointer is less saturated in each component than the others, then also the others are asymptotically stable provided that it satisfied condition (13) at layer H. Actually, since condition (13) is compounded of mutually independent conditions over the rows of $\boldsymbol{B}$, Theorem 3 can be extended to a set of fixed points $(\vec{d}(j)_H, \vec{\hat{d}}_O^{(x)})$, $j = 1, \ldots, S$, for the set of networks $X(j) \equiv (x, \vec{k}(j)^{(x)})$, satisfying the asymptotical stability condition (13). In this case, the fixed point $(\vec{d}(S+1)_H, \vec{\hat{d}}_O^{(x)})$, defined for the network $X(S+1) \equiv (x, \vec{k}(S+1)^{(x)})$, is asymptotically stable if the first condition of Theorem 3 is satisfied for each j and $\forall i,\ i = 1, \ldots, N_H,\ \exists j,\ s.t.\ f_i'(\vec{d}(S+1)_H) \ \leq f_i'(\vec{d}(j)_H)$.

If the stability condition is not satisfied at layer H, then the best we can state, without involving conditions more complex than the stability one, is the following theorem:

Theorem 4 *Given a fixed point* $(\vec{d}_H, \vec{\hat{d}}_O^{(x)})$ *for the network* $X \equiv (x, \vec{k}^{(x)})$ *satisfying the asymptotical stability condition (13), then the fixed point* $(\vec{d'}_H, \vec{\hat{d'}}_O^{(x)})$*, defined for the network* $X' \equiv (x, \vec{k'}^{(x)})$*, is asymptotically stable if:*

$\hat{\boldsymbol{E}}^{(x)} = \hat{\boldsymbol{D}}^{(x)^t} \hat{\boldsymbol{A}}^{(x)}$, *where* $\hat{\boldsymbol{A}}^{(x)}$ *is every diagonal matrix with components all of the same sign,* *and*

$$\forall i,\ i = 1, \ldots, N_H,\ f_i'(\vec{d'}_H) \ \le\ f_i'(\vec{d}_H),\ \forall j,\ j = 1, \ldots, \hat{N}_O^{(x)},\ f_j'(\vec{\hat{d'}}_O^{(x)}) \ \le\ f_j'(\vec{\hat{d}}_O^{(x)}).$$

Proof: The proof of this theorem can be obtained combining the observations made in the proofs of Theorems 3 and 4. The condition $\hat{\boldsymbol{E}}^{(x)} = \hat{\boldsymbol{D}}^{(x)^t} \hat{\boldsymbol{A}}^{(x)}$ guarantees that a decrease in the moduli of the diagonal matrices leads to a decrease in the moduli of the elements of the $\boldsymbol{B}$ matrix satisfying the stability condition. □

Actually, this theorem is more general than the previous one, since in this case we are considering every couple of fixed points of networks with the same structure but eventually different bias term at the hidden layer. Because of this generality, the conditions which must be satisfied tend to be more tight. An extended version of Theorem 4, considering a set of fixed points, can be easily proved as well.

6 Simulation

In this section, we give an example of graph encoding using the LRAAM model. When possible the theorems given in the previous sections are applied, under the hypothesis that each pattern in the final training set is very close to a fixed point of the system. Consequently, all the quantities involved by the conditions of the theorems are computed using this approximation. Therefore, the obtained results should be considered keeping in mind this aspect. In general, however, we found the obtained results to be more strict than the experimental ones. This is due to the over estimation introduced by the Gershgorin theorem of the modulus of the eigenvalues of $\boldsymbol{B}$.

The example involves the encoding of the labeled graph shown in Figure 5 using a $11-3-11$ **PA** network. The representations used for the labels are reported in Table I. The last two bits

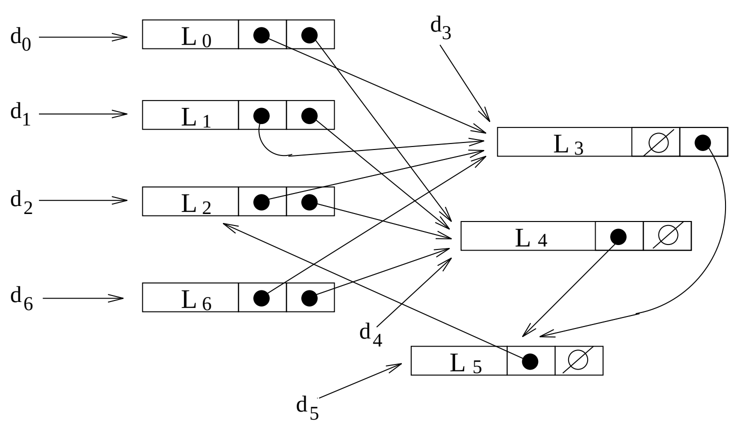

Figure 5: Labeled graph used for an experimental verification of the theoretical results.

of each label are used to represent the void condition for the pointer fields (-1 represents the void pointer condition).

It must be pointed out that the use of a part of the label to represent the void condition for the pointer fields is particularly efficient since the pointer fields are free to assume every configuration

Label	Code
L_0	-1 -1 -1 1 1
L_1	-1 -1 1 1 1
L_2	-1 1 -1 1 1
L_3	-1 1 1 -1 1
L_4	1 -1 -1 1 -1
L_5	1 -1 1 1 -1
L_6	1 1 -1 1 1

Table I: Representations for the labels of the graph shown in Figure 5.

when they are void, and this adds more degrees of freedom to the system. In order to avoid instabilities for the void pointers, their output activation at one epoch is used as input activation at the next epoch[4].

A standard backpropagation algorithm with momentum term and symmetrical sigmoidal units was used. In particular, the gains of the hidden units were set to 0.5, the gains for the output units representing the label were set to 2, and the gains for the remaining output units to 1. A discussion about this kind of distribution for the gains of the units can be found in [16]. The initial weights for the **PA** network were initialized with real numbers uniformly distributed in the interval [-0.6,0.6]. The learning parameter η for the weights was set to 0.07, the momentum μ to 0.5, and the learning was stopped after 4097 epochs[5], when the error for each output unit and for each pattern in the training set was below 0.05.

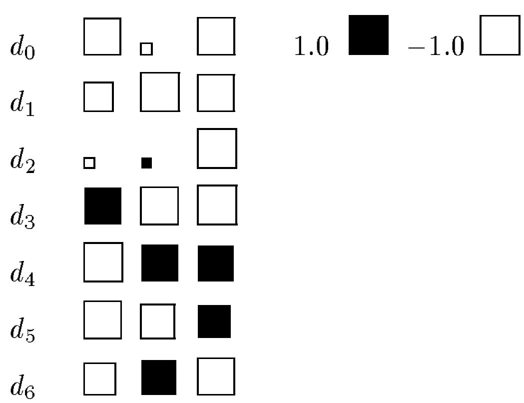

Figure 6: Distributed representations for the pointers of the labeled graph shown in Figure 5.

The distributed representations obtained for the pointers are graphically shown in Figure 6. These were obtained by presenting the corresponding input patterns to the input of the **PA** network and looking at the activation of the hidden units. The result of a cluster analysis over the

[4] Experimentation performed on this aspect showed fast convergence to different fixed points for different void pointers.

[5] The batch mode was used.

Cycle (d_2, d_3, d_5)	
Start	Result
d_2	1 1 1
d_3	1 1 0
d_5	1 1 1
Cycle (d_2, d_4, d_5)	
Start	Result
d_2	1 1 1
d_4	1 1 1
d_5	1 1 1

Table II: Results obtained by computing the asymptotical stability condition for the cycles of the graph shown in Figure 5.

representations is presented in Figure 7. It can be noted that the metric over the pointers reflects sufficiently well the relationships among the components of the graph.

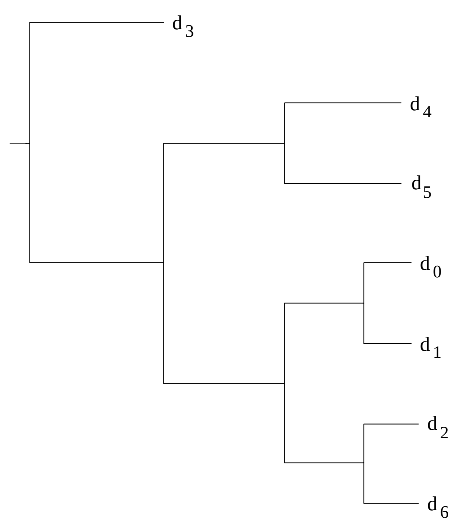

Figure 7: Cluster analysis of the distributed representations developed by the LRAAM for the pointers of the labeled graph shown in Figure 5.

The first test we performed on these representations was the application of Theorem 1 in order to ascertain the stability of the decoding process along the cycles (d_2, d_3, d_5) and (d_2, d_4, d_5). If these two cycles are asymptotically stable ones, the others given by a combination of them will also be. Observing that the pointers d_3 and d_4 have components almost saturated, a first positive guess on the asymptotical stability of the cycles can be made. The results of the computation of condition (13) for all possible starting points of each cycle are represented in an encoded form in Table II. The result is represented by three bits, one for each index of condition (13). If the value of the ith bit of the result is 1 it means that the condition is satisfied for the corresponding index, otherwise it is equal to 0. The starting pointer represents the first pointer decoded in the cycle.

From Table II it is clear that (d_2, d_3, d_5) and (d_2, d_4, d_5) are asymptotically stable cycles, since it suffices at least one starting pointer for which condition (13) is satisfied. Thus the first guess, given only on the basis of the degree of saturation of the pointers, was confirmed. The stability of the cycles was confirmed as well by direct inspection of the system behaviour during the decoding process.

In the second test we verified condition (13) over the functions $\boldsymbol{F}_E(\boldsymbol{F}_D())$ (test over layer H) and $\boldsymbol{F}_D(\boldsymbol{F}_E())$ (test over layer O). The results are reported in Table III, where we used the same convention of the previous table. In particular, the first column of the table reports the input pattern for the considered function, i.e., hidden activations for $\boldsymbol{F}_E(\boldsymbol{F}_D())$ and input activations (as represented in the final version of the training set) for $\boldsymbol{F}_D(\boldsymbol{F}_E())$.

FULL NETWORK ($x = 0$)	
Test over H	
Pattern	Result
d_0	1 1 1
d_1	1 1 1
d_2	1 1 1
d_3	1 1 1
d_4	1 1 1
d_5	1 1 0
d_6	1 1 1
Test over O	
Pattern	Result
(L'_0, d'_3, d'_4)	1 1 1 1 1 1 1 1 1 1 1
(L'_1, d'_3, d'_4)	1 1 1 1 1 1 1 1 1 1 1
(L'_2, d'_3, d'_4)	1 1 1 1 1 1 1 1 1 1 1
(L'_3, nil', d'_5)	1 1 1 1 1 1 0 1 1 0 1
(L'_4, d'_5, nil'')	1 1 1 1 1 1 0 0 0 1 0
(L'_5, d'_2, nil''')	1 1 1 1 1 0 0 1 0 1 0
(L'_6, d'_3, d'_4)	1 1 1 1 1 1 1 1 1 1 1

Table III: Results obtained by computing the asymptotical stability condition for the **CA** network.

From Table III we can state asymptotically stable all the fixed points defined on the **CA** network by the final version of the training set[6] but the fixed point $(d_5, [L'_5, d'_2, nil'''])$ for which condition (13) was satisfied neither at layer H nor at layer O. Actually, condition (13) was not satisfied at layer H because the sum of the modulus of the elements of the last row of the corresponding $\boldsymbol{B}$ matrix was 1.28. However, a direct inspection of the stability of the fixed point $(d_5, [L'_5, d'_2, nil'''])$ showed it was asymptotically stable as well as the others. Table III can also be used to apply Theorem 2. In particular, since the learned matrices $\boldsymbol{E}$ and $\boldsymbol{D}$ are such that $\forall x \in \{0, 1, 2, 3, 4, 5, 6\}$, $\nexists\ \hat{\boldsymbol{A}}^{(x)}$ s.t. $\hat{\boldsymbol{E}}^{(x)} = \hat{\boldsymbol{D}}^{(x)^t} \hat{\boldsymbol{A}}^{(x)}$, where $\hat{\boldsymbol{A}}^{(x)}$ is every diagonal matrix with components all of the same sign, we can use only the first condition of Theorem 2, i.e., that the stability condition is satisfied at

[6] Defined by (d_i, c_i), where c_i is the ith pattern in the final training set

layer O. In Figure 8 we have reported the inclusion graph regarding the structure of the networks derivable by the **CA** network. In the inclusion graph there is a node for each type of network structure derivable by the **CA** network, and an arrow from X to Y if $Y \subset X$. The **CA** network is represented by $(x, \vec{k}^{(x)})$, where $x = 0$ and $\vec{k}^{(0)} = \emptyset$. Networks with the same structure but different access key are identified by setting $\vec{k}^{(x)} = *$. According to Figure 8, Theorem 2 can be applied only

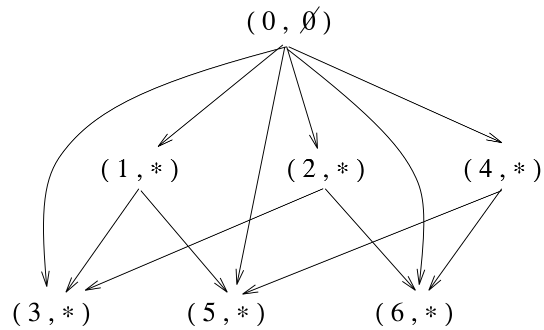

Figure 8: The inclusion graph for the networks derivable by the **CA** network.

to couples of networks for which there exists an arrow between them.

On the basis of Table III, it can be seen that the conditions of Theorem 2 are true for the first three fixed points and the last one. The consequence is that each subcomponent of the fixed points is asymptotically stable for each corresponding subnetwork, i.e., access procedure. Corollary 1 is instead useful if applied to the remaining fixed points, since it states that the fixed points (d_3, L'_3), (d_4, L'_4), and (d_5, L'_5) are asymptotically stable fixed points of the corresponding networks $(6, [nil', d'_5])$, $(6, [d'_5, nil''])$, and $(6, [d'_2, nil'''])$.

Let us now consider the results obtained for the networks $(1, *)$ (Table IV). From Table IV we can state that all the fixed points are asymptotically stable. Theorem 2 allows us also to state that (d_4, d'_5) is an asymptotically stable fixed point for the network $(5, [L'_4, nil''])$, and Corollary 1 that (d_3, d'_5) is an asymptotically stable fixed point for the network $(3, [L'_3, nil'])$.

The asymptotical stability of the others fixed points was verified by computing condition (13) at the layer H of the corresponding networks, and a direct inspection of the asymptotically stability of the fixed points confirmed the theoretical results. Thus, using Theorem 1, Theorem 2, and Corollary 1 we were able to foresee the stability properties of the fixed points. Only the fixed point (L'_5, d'_2, nil''') for the **CA** network escaped from the theoretical analysis.

The extended version of Theorem 3 was applied to the **CA** network considering the set of pointers $\{d_0, d_1, d_2\}$. In fact, the corresponding fixed points $(d_0, [L'_0, d'_3, d'_4])$, $(d_1, [L'_1, d'_3, d'_4])$, and $(d_2, [L'_2, d'_3, d'_4])$ satisfied condition (13) at layer H, d_2 had the less saturated first component, d_0 the second one, and d_1 the third one. Consequently $(d_6, [L'_6, d'_3, d'_4])$ was confirmed to be an asymptotically stable fixed point. As previously mentioned, $\boldsymbol{E}$ and $\boldsymbol{D}$ were such that it was not possible to apply Theorem 4.

7 Conclusions

In this report we have discussed some stability problems encountered in the LRAAM model. In particular, we have given sufficient conditions for the asymptotical stability of cycles in the structures encoded in the LRAAM, i.e., when the decoding of pointers belonging to a cycle can be performed

NETWORKS $(1,*)$		
Test over H		
Key	Pattern	Result
L'_0	d_0	1 1 1
L'_1	d_1	1 1 1
L'_2	d_2	1 1 1
L'_3	d_3	1 1 1
L'_4	d_4	1 1 1
L'_5	d_5	1 1 1
L'_6	d_6	1 1 1
Test over O		
Key	Pattern	Result
L'_0	(d'_3, d'_4)	1 1 1 1 1 1
L'_1	(d'_3, d'_4)	1 1 1 1 1 1
L'_2	(d'_3, d'_4)	1 1 1 1 1 1
L'_3	(nil', d'_5)	1 0 1 1 1 1
L'_4	(d'_5, nil'')	1 1 1 1 1 1
L'_5	(d'_2, nil''')	0 0 1 1 1 1
L'_6	(d'_3, d'_4)	1 1 1 1 1 1

Table IV: Results obtained by computing the asymptotical stability condition for the networks $(1,*)$.

without danger of loosing information. The same result can be applied with a few modifications to the asymptotical stability of fixed points of the **CA** network defined for the LRAAM. We have given as well sufficient conditions under which the property of asymptotical stability of a fixed point in one particular constrained version of the **CA** network can be extended to related fixed points of different constrained versions of the **CA** network. Experimental results have shown that the obtained theoretical results can be successfully applied to the trained network in order to foresee the reliability of the access procedure by content.

Several issues remain to be explored. First of all, the definition of a training procedure able to guarantee the asymptotical stability of the equilibria for every constrained version of the **CA** network and the asymptotical stability of the decoding process. Unfortunately, the procedure proposed by Atiya and Abu-Mostafa [1], in order to guarantee at least the asymptotical stability of fixed points for the unconstrained **CA** network, cannot be applied since it requires the saturation of the hidden units. Even if this requirement can be fulfilled in same cases by perturbing the solution obtained by the standard procedure, in general, the hidden representations need to be analog. Analog hidden representations are also preferable in view of the reduced representation framework, where the main goal is to obtain a continuos metric over the representations. Regarding this aspect, we believe there is a good possibility that the stability of the decoding procedure is obtained automatically by the standard procedure. This conjecture is partially supported by the fact that, under the hypotheses of perfect learning, linear output units, and representation of a cycle of length k in the same pointer field p, the eigenvalues of the matrix $(\boldsymbol{D}^{(p)})^k$, corresponding to the pointers of the cycle, must be equal to 1 (see [16]). If it is observed that, because of the linear units, $(\boldsymbol{D}^{(p)})^k$ corresponds to the Jacobian matrix of the decoding function along the cycle, and that in the nonlinear case there is a strong decay component in the Jacobian matrix given by the first derivative of the sigmoidal units, then the conjecture seems to be reasonable. Till now, we have never encountered an unstable cycle in our simulations.

Another research issue is the definition of the shape of the basin of attraction of the equilibria of the **CA** network. The next step will be the attempt to perform an analysis of a basin of attraction by the technique used by Michel and Farrell in [8]. Moreover, in order to improve the shape of the domain of attraction, an adaptation of the *unlearning* method exploited by Keeler [6] for binary Hopfield networks will be tested.

Spurious equilibria are not frequent in the unconstrained **CA** network. This can be a consequence of the fact that the fixed points are not completely defined by the user, since the actual value for the pointers is generated by the network. This claim needs to be formally demonstrated as well. Related to this aspect is also the observation that the capacity of the LRAAM model seems superior with respect to the results obtained by Atiya and Abu-Mostafa. In fact, in our example, a network with 3 hidden units was able to store 7 asymptotically stable fixed points. In any case, spurious attractors can be eliminated by introducing terminal attractors [18].

In conclusion, complex structures are encoded by an LRAAM in such a way that both distributed reduced representations and access by content can be exploited. Moreover, theoretical analysis on the asymptotical stability of equilibria can be performed in order to decide which component of the structure can be safely accessed by content. More work needs to be done in order to assess the potentialities of the model.

References

[1] A. Atiya and Y. S. Abu-Mostafa. An analog feedback associative memory. *IEEE Transaction on Neural Networks*, 4:117–126, 1993.

[2] J. A. Farrell and A. N. Michel. A synthesis procedure for Hopfield's continuous-time associative memory. *IEEE Transaction on Circuits and Systems*, 37:877–884, 1990.

[3] A. Guez, V. Protopopsecu, and J. Barhen. On the stability, storage capacity, and design of nonlinear continuous neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 18:80–87, 1988.

[4] G. E. Hinton. Mapping part-whole hierarchies into connectionist networks. *Artificial Intelligence*, 46:47–75, 1990.

[5] J. J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. In *Proc. Natl. Acad. Sci.*, 81:3088–3092, 1984.

[6] J. D. Keeler. Basins of attraction of neural network models. In *AIP Conference Proceedings*, 151:259–265, 1986.

[7] B. Kosko. *Neural Networks and Fuzzy Systems*. Prentice Hall, 1992.

[8] A. N. Michel, J. A. Farrell, and W. Porod. Qualitative analysis of neural networks. *IEEE Transaction on Circuits and Systems*, 36:229–243, 1989.

[9] F. J. Pineda. Dynamics and architecture for neural computation. *Journal of Complexity*, 4:216–245, 1988.

[10] T. Plate. Holographic reduced representations. Technical Report CRG-TR-91-1, Department of Computer Science, University of Toronto, 1991.

[11] J. B. Pollack. *Implications of Recursive Distributed Representations*, pages 527–536. Advances in Neural Information Processing Systems I. San Mateo: Morgan Kaufmann, 1989.

[12] J. B. Pollack. Recursive distributed representations. *Artificial Intelligence*, 46(1-2):77–106, 1990.

[13] D. E. Rumelhart and J. L. McClelland. *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. MIT Press, 1986.

[14] F. M. A. Salam, Y. Wang, and M-R. Choi. On the analysis of dynamic feedback neural nets. *IEEE Transaction on Circuits and Systems*, 38:196–201, 1991.

[15] P. Smolensky. Tensor product variable binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, 46:159–216, 1990.

[16] A. Sperduti. Labeling RAAM. Technical Report 93-029, International Computer Science Institute, 1993.

[17] D. S. Touretzky. Boltzcons: Dynamic symbol structures in a connectionist network. *Artificial Intellicence*, 46:5–46, 1990.

[18] M. Zak. Terminal attractors in neural networks. *Neural Networks*, 2:259–274, 1989.