Storage of Linguistic Information in a Continuous Classifying Associative Memory

Antonio B. Bailón,* Miguel Delgado, Waldo Fajardo
Dpto. Ciencias de la Computación e Inteligencia Artificial, E.T.S. Ingeniería Informática—Universidad de Granada, Avda. Andalucía 38, 18071 Granada, Spain

In this article, we analyze the use of the continuous classifying associative memory (CCLAM) to store linguistic information. Freedom in the choice of the functions that control the operation of the CCLAM equip this memory with the capacity to adapt to different information storage and recovery needs. We begin with the problem of storing linguistic terms by memorizing the patterns formed by the degrees of compatibility with these terms. After that, the problem of storing linguistic rules is discussed. Let us remark that in these cases not a single CCLAM is used, but rather a set of them connected in suitable structured ways. © 2002 Wiley Periodicals, Inc.

1. INTRODUCTION

There is one serious problem in associative memories: their low storage capacity caused by the appearance of spurious patterns; that is, patterns that do not correspond to any of the ones stored.

The classifying associative memory (CLAM)[1,2,3] solves the problem of capacity by avoiding the factors that limit it: spurious states and data dependency. It does so by not limiting the type of data that can be stored, at the same time as preventing interference occurring between any of the stored patterns that would produce spurious patterns.

If we reflect on the type of information that human beings normally memorize or use in their reasoning, we find that they manage a large quantity of imprecise concepts. Although this might seem like a limitation, it enables them to successfully face up to extremely complicated tasks in which computers, although a lot more precise than humans, fail. This inverse relationship and complexity is expressed in Zadeh's incompatibility principle.[8]

If we wish to model complex systems, we need to reduce the precision with which they must be dealt with. Consequently, memories capable of storing imprecisely expressed information are necessary.

*To whom all correspondence should be addressed; e-mail: bailon@decsai.ugr.es.

INTERNATIONAL JOURNAL OF INTELLIGENT SYSTEMS, VOL. 17, 609–627 (2002)

Published online in Wiley InterScience (www.interscience.wiley.com). • DOI: 10.002/int.10041

Fuzzy associative memories are associative memories that enable us to store and recover imprecise information. They represent a very important element of automatic information treatment systems due to their behavior, which is similar to that of human memory.

The continuous classifying associative memory (CCLAM)[4,5] retains the high storage capacity of the discrete CLAM and enables us to store patterns with continuous components in the interval [0, 1].

We will use the CCLAM to store imprecise information expressed in linguistic terms using patterns formed by the degrees of compatibility with these terms.

Our work is divided into the following sections:

- We begin with a short description of the continuous classifying associative memory, showing its topology and the learning and recall processes.
- Next we show how to use the CCLAM to store linguistic information:
  - Storage of labels of linguistic variables
  - Storage of fuzzy rules
- Finally we show a conclusion set obtained from this article and the bibliography used.

2. CONTINUOUS CLASSIFYING ASSOCIATIVE MEMORY

The CCLAM has been introduced and thoroughly discussed.[4,5] Here we will only summarize its main characteristics in order to make this article readable. The CCLAM stores special arbitrary patterns that will have continuous components in the interval [0, 1]. The topology of the CCLAM (See Figure 1) differs from the CLAM in that all the processing elements (PEs)[4] are continuous instead of bipolar

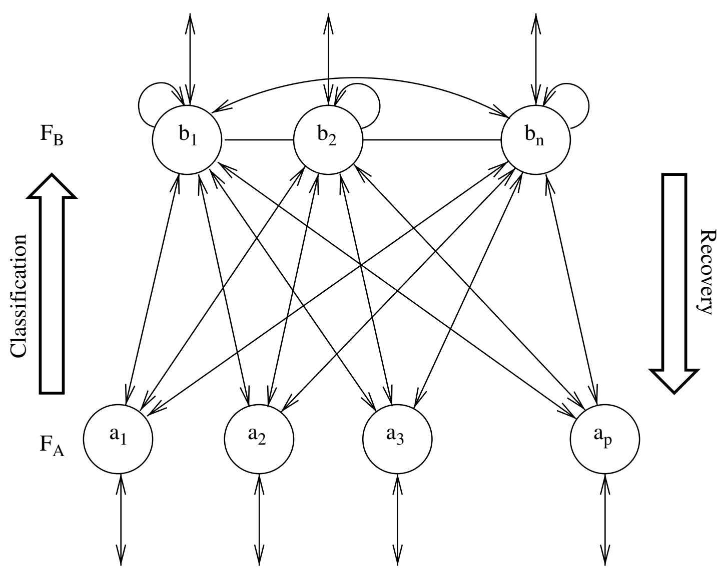

Figure 1. Continuous classifying associative memory (CCLAM).

and binary; they have no threshold and their activation function is:

$$f(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } 0 \leq x \leq 1 \\ 1 & \text{if } x > 1 \end{cases} \quad (1)$$

In layer F_A there are as many PEs as components of the stored patterns. The input patterns will be presented in this layer and the stored patterns will be recovered. The PEs of layer F_B represent the stored patterns. Their number is variable and changes dynamically with learning, so that there will always be as many PEs in layer F_B as there are stored patterns. Their activation state indicates the degree to which the associated patterns are recognized.

Layer F_B is competitive. The weights of the competition signals between the PEs of layer F_B are represented in the matrix $\beta_{n\times n}$, with n being the number of stored patterns and β_{ij} being the weight of the existing connection from b_i to b_j. The winning element is the one for which the activation state Y_i verifies $Y_i \geq Y_j \beta_{ji}\ \forall j$. If there is more than one winning element, one of them will be selected following an established criterion and the others will be deactivated.

There are two different processes in the CCLAM operation: classification and recovery. The weights and functions used in each process might be different, which is why they will be distinguished using a superscript.

2.1. Learning

In the learning process, the memorization of a pattern entails the creation of a new PE in F_B and the appropriate adjustment of the weights of the new connections that are made towards it.

The learning process has the following steps:

(1) Arrival at F_A of a pattern that we want to memorize.
(2) Verification that the pattern was not previously memorized.
(3) Creation of a new PE in F_B associated with the pattern.
(4) Adjustment of the competition weights of layer F_B.
(5) Adjustment of the weights of F_A towards the new PE. When the pattern $X = (x_1, \ldots, x_p)$ is stored, the weights of the PE connections of layer F_A towards the new PE b_p of layer F_B are assigned the value $m_{pt} = x_i$.

2.2. Classification

In the classification process, the PEs send activation signals from the layer F_A towards layer F_B. Given an input pattern presented in the layer F_A, we can obtain activation states in layer F_B that represent the similarity that exists between the input pattern and each of the stored patterns.

There are two steps to the classification process:

(1) A pattern $X = (x_1, \ldots, x_p)$ arrives at layer F_A.

(2) The signal spreads towards layer F_B so that the PE b_i receives the activation signal:
   (a) $y_i = \Psi^C\{\Upsilon^C(x_i, m_{ij}^C)\}_{j=1\ldots p}$.
   (b) Υ^C and Ψ^C are functions bounded to the interval [0, 1].

(3) The PEs of layer F_B compete against each other with the intensity reflected in the weights matrix β^C.

(4) We obtain the result of the classification process in layer F_B. The activation state obtained in the PEs of layer F_B indicates the degree to which the input pattern is classified for each of the classes represented by the stored patterns.

2.3. Recovery

In the recovery process, the information flow is contrary to that of the classification process. In layer F_B, an input pattern is presented that represents the degree to which recovery involves each of the stored patterns. The recovery has the following steps:

(1) A pattern $Y = (y_1, \ldots, y_n)$ arrives at layer F_B.

(2) The PEs of layer F_B compete against each other with the weights matrix β^R. The new activation states form the pattern $Y' = (y'_1, \ldots, y'_n)$.

(3) The signal spreads from layer F_B towards layer F_A so that the PE a_i receives the activation signal:
   (a) $x_i = \Psi^R\{\Upsilon^R(y'_j, m_{ji}^R)\}_{j=1\ldots n}$.
   (b) Υ^R and Ψ^R are functions bounded to the interval [0, 1].

(4) We obtain the pattern resulting from the recovery process in layer F_A. This is either one of the stored patterns or a combination of them.

3. STORAGE OF LABELS OF LINGUISTIC VARIABLES

If we want to store patterns that code different characteristics of a set of objects, we need to use several linguistic variables to describe them. In order to store information about several linguistic variables, the continuous classifying associative memory will be used as a structural element so that we will have a CCLAM for each linguistic variable, as well as another CCLAM that will unify all the information and receive (in layer F_A) the signals obtained in layer F_B of the CCLAMs associated with the linguistic variables.

Each CCLAM associated with a linguistic variable will have as many PEs in F_A as there are linguistic variables defined on the variable, and as many PEs in F_B as there are memorized patterns. The unifying CCLAM will have as many PEs in F_B as there are memorized patterns. For each linguistic variable in F_A, there will be as many PEs as there are stored patterns.

For each linguistic variable we will compute the similarity that exists between the input coded value and the stored value, and the unifying CCLAM will combine all the information in order to determine which pattern will be recovered. The behavior of the system in the classification and recovery phases of the CCLAMs associated with linguistic variables, and in the unifying CCLAM, will be shown in Figure 2.

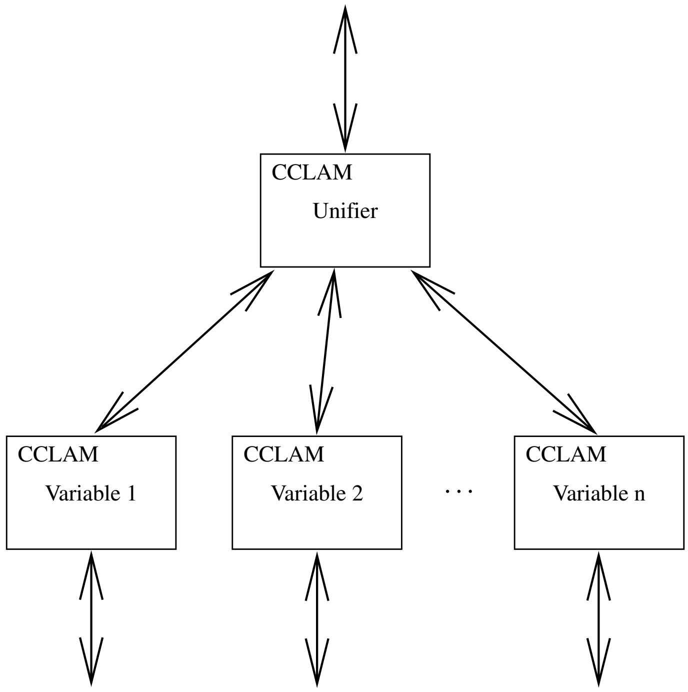

Figure 2. Storage of linguistic labels using CCLAM.

3.1. Classification

In the classification process the pattern with the greatest similarity to the input pattern is determined and recovered. This entire process in divided into two phases.

The first phase occurs in the CCLAMs associated with the linguistic variables. In each, the similarity that exists between the input pattern and each of the stored patterns is determined.

In the second phase, developed in the unifier, from the individual similarities obtained in the first phase, we obtain the overall similarity that exists between the input pattern and the memorized patterns. The pattern with which it shares the greatest similarity will be the one selected.

3.1.1. CCLAM Associated with Linguistic Variables

The comparison made by using Υ^C and Ψ^C will show us the degree of similarity between the input pattern and the stored patterns.

The Υ^C function indicates the similarity that exists between the individual components. For that reason we propose to use $\Upsilon^C(x, y) = 1 - |x - y|$.

The function Ψ^C is responsible for combining the particular information obtained from the components and that's why we propose the arithmetic mean of their similarities.

The F_B layer receives the mean of the degrees of similarity that exist between the components of the input pattern and those of the stored patterns. The PE with the greatest activation state will be the one that represents the stored pattern closest to the input pattern.

There will be no competition in layer F_B, because the aim of the competition is to select the PE that represents the closest pattern and this task is carried out by the unifier after receiving the information from the rest of the CCLAM.

3.1.2. Unifier

The CCLAM that unifies the information receives in its F_A layer the state of the PEs of layer F_B of the CCLAM associated with the linguistic variables.

Let X be the PE of the F_A layer that receives its activation state of the element representing the jth pattern of the CCLAM associated with the ith linguistic variable. Let Y be the PE of the F_B layer associated to the kth pattern. The weight of the connection made between X and Y is:

$$w_{ijk} = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

That is to say, the PEs associated with the same pattern have a weight of one, whereas the rest have a weight of zero. Consequently, each PE of F_B only receives information from the PEs of F_A associated with the same pattern.

For the propagation of the signals from layer F_A to layer F_B we will configure the CCLAM with:

- The function $\Upsilon^C(x, y) = xy$, which ensures that each PE of the F_B layer only receives signals from the elements associated with the same pattern.
- The arithmetic mean used as the Ψ^C function to compute the global similarity that exists between the input pattern and the corresponding stored pattern.
- A competition in which the winning element takes the value of one, which ensures that the only active PE of F_B is the one that represents the stored pattern that is most similar to the input pattern. For this, the competition weights are assigned the value of one so that the same importance is given to all the PEs.

3.2. Recovery

The recovery process begins once the stored pattern to be recovered has been determined. The unifier will only transmit the selection to the CCLAMs associated with the variables so that these can recover the information of the pattern selected.

3.2.1. Unifier

During the recovery, the unifier transmits the selection made to the CCLAMs associated with the variables.

The weights of the connections between the F_A and F_B layers are the same as those established in the classification phase; that is to say the PEs associated with the same pattern are linked with a connection with a weight of one, whereas the others are linked by connections with a weight of zero.

Competition is nullified because the elements have already competed at the end of the classification phase and only one of them is active. The competition weights are therefore set to zero: $\beta_{ij}^R = 0 \;\; \forall i, j$.

Using $\Upsilon^R(x, y) = xy$, each element of F_A will only receive information from the PEs associated with the same pattern. In this way, the elements of F_A associated with the selected pattern will receive an active signal from F_B. The elements associated with unselected patterns will receive all the signals at zero.

Using $\Psi^R(x_1, \ldots, x_n) = \max\{x_1, \ldots, x_n\}$, only the elements of F_A associated with the pattern to be recovered remain active.

3.2.2. CCLAM Associated with Linguistic Variables

Each CCLAM associated with a linguistic variable receives a signal from the unifier that leaves a single active PE in its F_B layer. This element, whose state is one, is the one that represents the pattern to be recovered. It is therefore not necessary to establish a competition between the PEs of F_B so that all the weights will be set to zero: $\beta_{ij}^R = 0 \;\; \forall i, j$.

The functions Υ^R and Ψ^R that will be used are the minimum and maximum, respectively. This ensures that all the inactive PEs of F_B send zero activation signals, and that the final activation state of the elements of F_A is the one received from the element that represents the pattern that will be recovered.

3.3. Example

We are going to store the height, weight, and age of a set of individuals. These attributes will be expressed in linguistic terms.

Let us take the linguistic variable HEIGHT. In this linguistic variable we are going to use the linguistic labels VERY SMALL, SMALL, NORMAL, TALL, and VERY TALL. The linguistic labels VERY LIGHT, LIGHT, NORMAL, HEAVY, and VERY HEAVY will be used with the linguistic variable WEIGHT. Finally, we will take the linguistic variable AGE, with which the linguistic labels CHILDHOOD, PUBERTY, YOUTH, MIDDLE AGE, THIRD AGE, and OLD AGE will be used.

The patterns that we wish to store correspond to three people:

	HEIGHT				
	VS	S	N	T	VT
Toni	0	0	0	0.5	0.5
David	0	0	0.9	0.1	0
Ana	0	0	0.2	0.8	0

	WEIGHT				
	VL	L	N	H	VH
Toni	0	0	0	0.9	0.1
David	0	0.3	0.7	0	0
Ana	0	0	0.6	0.4	0

	AGE					
	C	P	Y	MA	TA	OA
Toni	0	0	0.8	0.2	0	0
David	0	0	0	0.4	0.6	0
Ana	0	0.9	0.1	0	0	0

The learning process will produce the following weights matrixes:

$$M_{HEIGHT} = \begin{pmatrix} 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0.9 & 0.1 & 0 \\ 0 & 0 & 0.2 & 0.8 & 0 \end{pmatrix} \tag{3}$$

$$M_{WEIGHT} = \begin{pmatrix} 0 & 0 & 0 & 0.9 & 0.1 \\ 0 & 0.3 & 0.7 & 0 & 0 \\ 0 & 0 & 0.6 & 0.4 & 0 \end{pmatrix} \tag{4}$$

$$M_{AGE} = \begin{pmatrix} 0 & 0 & 0.8 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0.4 & 0.6 & 0 \\ 0 & 0.9 & 0.1 & 0 & 0 & 0 \end{pmatrix} \tag{5}$$

$$M_{UNIF} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{6}$$

Let us suppose that the input is $(\hat{H}, \hat{W}, \hat{A})$, where $\hat{H}$ is the subpattern that expresses the height, $\hat{W}$ expresses the weight, and $\hat{A}$ expresses the age, assessed by:

$$\begin{aligned} \hat{H} &= (h_{VS}, h_S, h_N, h_T, h_{VT}) = (0, 0, 1, 0, 0) \\ \hat{W} &= (p_{VL}, p_L, p_N, p_H, p_{VH}) = (0, 0, 0.5, 0.5, 0) \\ \hat{A} &= (e_C, e_P, e_Y, e_{MA}, e_{TA}, e_{OA}) = (0, 1, 0, 0, 0, 0) \end{aligned}$$

The input pattern is therefore:

$$(0, 0, 1, 0, 0, \quad 0, 0, 0.5, 0.5, 0, \quad 0, 1, 0, 0, 0, 0)$$

The functions used to transmit the information from layer F_A towards F_B in the CCLAMs associated with the linguistic variables are $\Upsilon^C(x, y) = 1 - |x - y|$ and the Ψ^C arithmetic mean.

The classification process that is carried out in the CCLAM associated with the height produces the results shown in the following table. In the cell corresponding to row S and column T, the result of the function $\Upsilon^C(h_T, m^C_{T,S})$ appears. At the end of the row, the value returned by Ψ^S is shown when applied to all the values of this row.

Υ^C	VS	S	N	T	VT	Ψ^C
Toni	1	1	0	0.5	0.5	0.6
David	1	1	0.9	0.9	1	0.96
Ana	1	1	0.2	0.2	1	0.68

The values received in layer F_B are $Y_{Toni} = 0.6$, $Y_{David} = 0.96$, and $Y_{Ana} = 0.68$.

All the competition weights have a value of zero so that all the PEs of layer F_B retain their state.

The classification process that is carried out in the CCLAM associated with the weight produces the results shown in the following table. In the cell corresponding to row S and column T, the result of the function $\Upsilon^C(p_T, m^C_{T,S})$ appears. At the end of the row, the value returned by Ψ^C is shown when applied to all the values of this row.

Υ^C	VL	L	N	H	VH	Ψ^C
Toni	1	1	0.5	0.6	0	0.62
David	1	0.7	0.8	0.5	1	0.8
Ana	1	1	0.9	0.9	1	0.96

The values received in layer F_B are $Y_{Toni} = 0.62$, $Y_{David} = 0.8$, and $Y_{Ana} = 0.96$.

All the competition weights have a value of zero so that all the PEs of layer F_B retain their state.

The classification process that is carried out in the CCLAM associated with the age produces the results shown in the following table. In the cell corresponding to row S and column T, the result of the function $\Upsilon^C(e_T, m^C_{T,S})$ appears. At the end of the row, the value returned by Ψ^C is shown when applied to all the values of this row.

Υ^C	C	P	Y	MA	TA	OA	Ψ^C
Toni	1	0	0.2	0.8	1	1	0.67
David	1	0	1	0.6	0.4	1	0.67
Ana	1	0.9	0.9	1	1	1	0.97

The values received in layer F_B are $Y_{Toni} = 0.67$, $Y_{David} = 0.67$, and $Y_{Ana} = 0.97$.

All the competition weights have a value of zero so that all the PEs of layer F_B retain their state.

The unifying CCLAM receives the signal from layers F_B of the CCLAMs associated with the height, weight, and age. When this signal is spread towards F_B using the functions Υ^C and Ψ^C, the signals received are $Y_{Toni} = 0.21$, $Y_{David} = 0.27$, and $Y_{Ana} = 0.29$.

All the competition weights have a value of one. The element that wins the competition passes to state one while the others remain deactivated. The state of the elements of F_B after the competition will therefore be $Y_{Toni} = 0$, $Y_{David} = 0$, and $Y_{Ana} = 1$. This means that the stored pattern closest to the input pattern corresponds to Ana.

During the recovery, the unifier transmits the selection carried out to the CCLAMs associated with the variables.

The competition is nullified once $\beta_{ij}^R = 0 \ \forall i, j$ has been established, which is why the PEs of F_B retain their state.

The weights of the connections that are established between layers F_A and F_B are the same as those established in the classification phase and $\Upsilon^R(x, y) = xy$. For that reason the PEs associated with the same pattern are those that receive non-zero signals. In this way, the signal received in layer F_B of the CCLAM associated with the height is $Y_{Toni} = 0$, $Y_{David} = 0$, and $Y_{Ana} = 1$, the signal received in the CCLAM associated with the weight is $Y_{Toni} = 0$, $Y_{David} = 0$, and $Y_{Ana} = 1$, and the signal received in the CCLAM associated with the age is $Y_{Toni} = 0$, $Y_{David} = 0$, and $Y_{Ana} = 1$.

In the recovery process there is no competition in the CCLAMs associated with linguistic variables; that is, the weights have been adjusted so that the competition does not modify any activation state. Because the weights are $\beta_{ij}^R = 0 \ \forall i, j$, no element changes state.

The signals that propagate from F_B towards F_A in the CCLAMs of the height, weight, and age are:

HEIGHT				
VS	S	N	T	VT
0	0	0.2	0.8	0

WEIGHT				
VL	L	N	H	VH
0	0	0.6	0.4	0

AGE					
C	P	Y	MA	TA	OA
0	0.9	0.1	0	0	0

Therefore, the pattern that is most similar to the input pattern that is recovered is:

$$(0, 0, 0.2, 0.8, 0, \quad 0, 0, 0.6, 0.4, 0, \quad 0, 0.9, 0.1, 0, 0, 0) \tag{7}$$

This is the one corresponding to Ana's stored information.

4. STORAGE OF FUZZY RULES

When the type of information that we need to store is made up of fuzzy rules, we can expect a different behavior than that obtained when only object attributes are stored.

The storage of rules entails memorizing pairs of patterns (antecedent, consequent), whereas the storage of object information entails memorizing individual patterns.

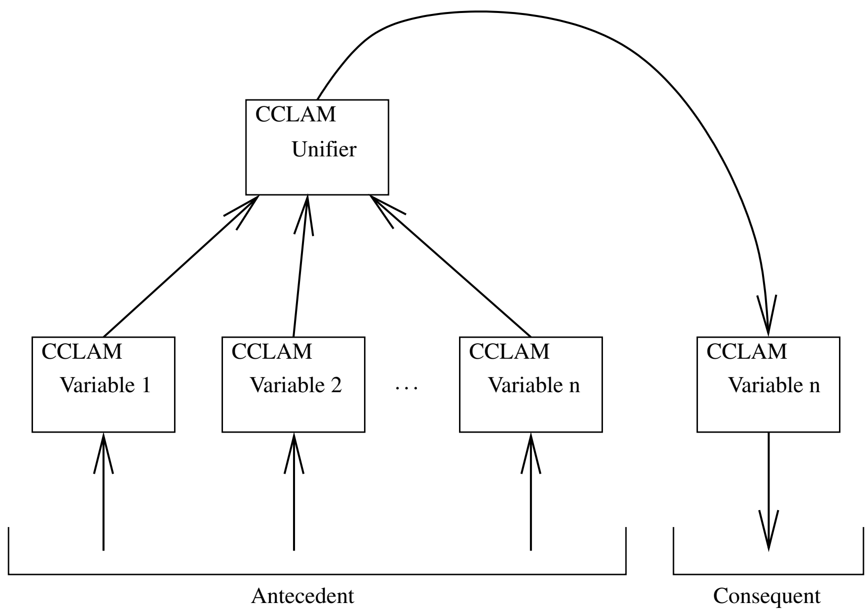

Figure 3. Storage of rules using CCLAM.

When we store object attributes, we want the system to recover the stored pattern that is closest to the input pattern. When we store rules of form (antecedent, consequent), when a pattern is provided as the antecedent, we can expect to obtain a consequent pattern, which need not necessarily be one of the stored patterns, but rather the result of the application of all the rules that are fired.

In order to store rules for inference purposes, we have a CCLAM for each of the linguistic variables of the antecedent and another CCLAM for the consequent. The memories associated with the antecedent will receive a pattern as the input that will determine the state of the system for which the rules will be applied. Each of these memories will calculate the degree to which the value of the linguistic variable that it represents causes the firing of each of the rules. An additional CCLAM will unify all the information, finding the total degree to which each rule is fired. This information is then sent to the memory, which stores the consequents so that the appropriate combination of these may be recovered.

We can look at this outline in more detail. Let us suppose the linguistic variables $A_1, A_2, \ldots, A_n$ form the antecedent and the linguistic variable C forms the consequent.

Lets assume that the term sets are $\{t_{i1}, t_{i2}, \ldots, t_{iq_i}\}$ for A_i, $i = 1, \ldots, n$, and $\{c_1, \ldots, c_m\}$ for C.

Then the set of rules to be stored is $\{R_1, R_2, \ldots, R_p\}$, where each rule is of the form *IF* a_1 *AND* a_2 *AND* ... *AND* a_n *THEN* b, with:

$$\begin{aligned} a_j &\in \{t_{j1}, \ldots, t_{jq_j}\} \\ b &\in \{c_1, \ldots, c_m\} \end{aligned} \tag{8}$$

The CCLAM system that will store the set of rules is shown in Figure 4.

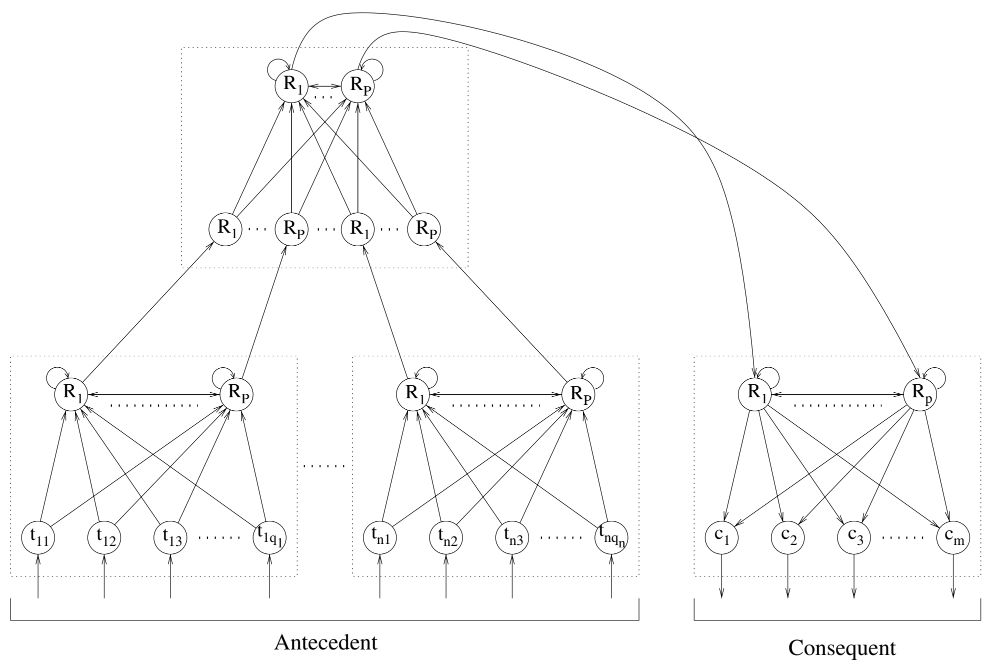

Figure 4. CCLAM that stores the set of rules.

Below we will specify the behavior of the CCLAM associated with both the antecedent as well as the consequent.

4.1. Antecedent

The CCLAMs associated with the linguistic variables of the antecedent and the unifying CCLAM make up the antecedent. In these memories, only the classification process will be used; that is, the information will only flow from layers F_A towards layer F_B.

4.1.1. CCLAM Associated with a Linguistic Variable of the Antecedent

In the memories associated with the linguistic variables of the antecedent, the degree to which each of the stored rules is applicable will be determined.

Since the rules are encoded in the weights of the connections between the layers F_A and F_B, a t-norm will be used as the Υ^C function. In this way, so that PE of F_B receives a non-zero signal, it is necessary for the input pattern to activate a PE of F_A, which is part of the rule represented by the element of F_B.

The function Ψ^C that is used is a t-conorm, so that only the elements that received a non-zero signal from F_A remain active in F_B. The activation state that results in the PE of F_B reflects the degree of fulfillment of each of the stored rules.

The competition of F_B will be nullified because we want to know all the rules that will be fired and not just the one that will to the greatest extent. Therefore, the weights of the competition connections will be $\beta_{ij}^C = 0 \ \forall i, j$.

4.1.2. Unifier

The unifying CCLAM will have as many PEs as memorized rules in F_B, and there will be as many PEs as rules for each linguistic variable of the antecedent in F_A. The weights of the connections that are established between layers F_A and F_B are such that the PEs associated with the same rule are united by connections with a weight of zero, whereas the remainder are united by connections with a weight of one. This allocation of weights is due to the fact that the value of one is the neutral element of the t-norms, which will be useful in the spreading of information from F_A to F_B.

The degree to which a rule is fired is calculated by applying a t-norm to the extent to which the antecedent variables are fulfilled. For this, a t-norm will be used, normally the minimum, as the function Ψ^C. This t-norm will be applied to information from the PEs of F_A.

We want only the activation states of the PEs associated with the same rule to be considered. To achieve this, it is necessary to receive the neutral element of the t-norm from the PEs associated with different rules. Given the weights of the connections between the layers F_A and F_B, the selected Υ^C function ought to be a t-conorm, for example the maximum. The PEs associated with the different rules, because they are connected with unitary weights, send the neutral element of the t-norm towards F_B; since the elements associated with the same rules are connected with zero weights and these are the neutral elements of the t-norm, they send their own activation state towards F_B.

We eliminate the competition from F_B because we are not searching for a winning rule but rather the degree to which each of them is fired. The competition weights will therefore be $\beta_{ij}^C = 0 \ \forall i, j$.

For their activation state, the PEs of F_B finally have the degree to which the rule that representing each of them must be fired.

4.2. Consequent

The CCLAM associated with the consequent receives the activation state of the PEs of layer F_B from the unifier as the input. That is, each PE takes the degree to which the rule to which it is associated as the activation state. The result we expect to obtain is that obtained from the application of all the rules that must be fired.

We do not want there to be competition between the PEs of F_B because we are not searching for a winning element. All the competition weights are therefore nullified: $\beta_{ij}^R = 0 \ \forall i, j$.

As the Υ^R function, a t-norm will be selected. In this way, the rules that are not fired, that is, those whose representing elements have a zero activation state, will send the value zero (which is the neutral value of the t-conorms) towards layer F_A. Selecting one t-conorm as the function Ψ^R, we obtain the result of the application of all the rules that are fired in F_A.

4.3. Example

Let us suppose that we want to store the set of rules that controls the braking of a moving object by applying a force that depends on the object's position and speed.

Let us suppose that we express the position by means of the linguistic variable POSITION with labels HIGH NEGATIVE, LOW NEGATIVE, ZERO, LOW POSITIVE, and HIGH POSITIVE.

The speed will be expressed by means of the linguistic variable SPEED with labels NEGATIVE, ZERO, and POSITIVE.

The response offered by the system for the braking of the moving object will correspond to the force that necessary to apply to it. This will be represented by the variable FORCE with labels HIGH NEGATIVE, LOW NEGATIVE, ZERO, LOW POSITIVE, and HIGH POSITIVE.

The set of rules that characterizes the system is:

(1) If high positive position and zero speed, then high negative force
(2) If low positive position and positive speed, then low negative force
(3) If low positive position and negative speed, then zero force
(4) If high negative position and zero speed, then high positive force
(5) If low negative position and negative speed, then low positive force
(6) If low negative position and positive speed, then zero force
(7) If zero position and zero speed, then zero force

The set of CCLAMs that will store the rules will memorize the patterns (P, S, F), where P and S represent the position and speed of the moving object, respectively, and F is the pattern that represents the force to apply to stop it. The coding of the rules will produce the following set of patterns:

	Position					Speed			Force				
Reg.	HN	LN	ZE	LP	HP	NE	CE	PO	HN	LN	ZE	LP	HP
1	0	0	0	0	1	0	1	0	1	0	0	0	0
2	0	0	0	1	0	0	0	1	0	1	0	0	0
3	0	0	0	1	0	1	0	0	0	0	1	0	0
4	1	0	0	0	0	0	1	0	0	0	0	0	1
5	0	1	0	0	0	1	0	0	0	0	0	1	0
6	0	1	0	0	0	0	0	1	0	0	1	0	0
7	0	0	1	0	0	0	1	0	0	0	1	0	0

In order to memorize the rules, we will use four CCLAMs. Two of them will be used to store the antecedents of the rules, that is, the values of position and speed. Another CCLAM will unite the results of these two memories, and the information obtained will be sent to a fourth and last CCLAM responsible for recovering the information relating to the force to be applied.

After the learning process, the CCLAM associated with the position will have the following weights matrix:

$$M = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{9}$$

The weights matrix of the CCLAM associated with the speed is:

$$M = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \tag{10}$$

The information from the memories associated with position and speed is spread towards the unifying CCLAM. This has fourteen PEs in F_A (for each of the two linguistic variables of the antecedent) and seven elements in F_B. The weights matrix is:

$$M = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \tag{11}$$

The CCLAMs associated with position and speed and the unifying CCLAM are those involved in the classification process. From an input pattern specifying the position and speed of the moving object, we obtain the degree to which each of the system rules is fired. This information is sent to a CCLAM that recovers a pattern that represents the force that the system of rules has determined must be applied. The weights matrix of this CCLAM associated with the force is:

$$M = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix} \tag{12}$$

From these ideas it is easy to show that the set of CCLAMs used to memorize the

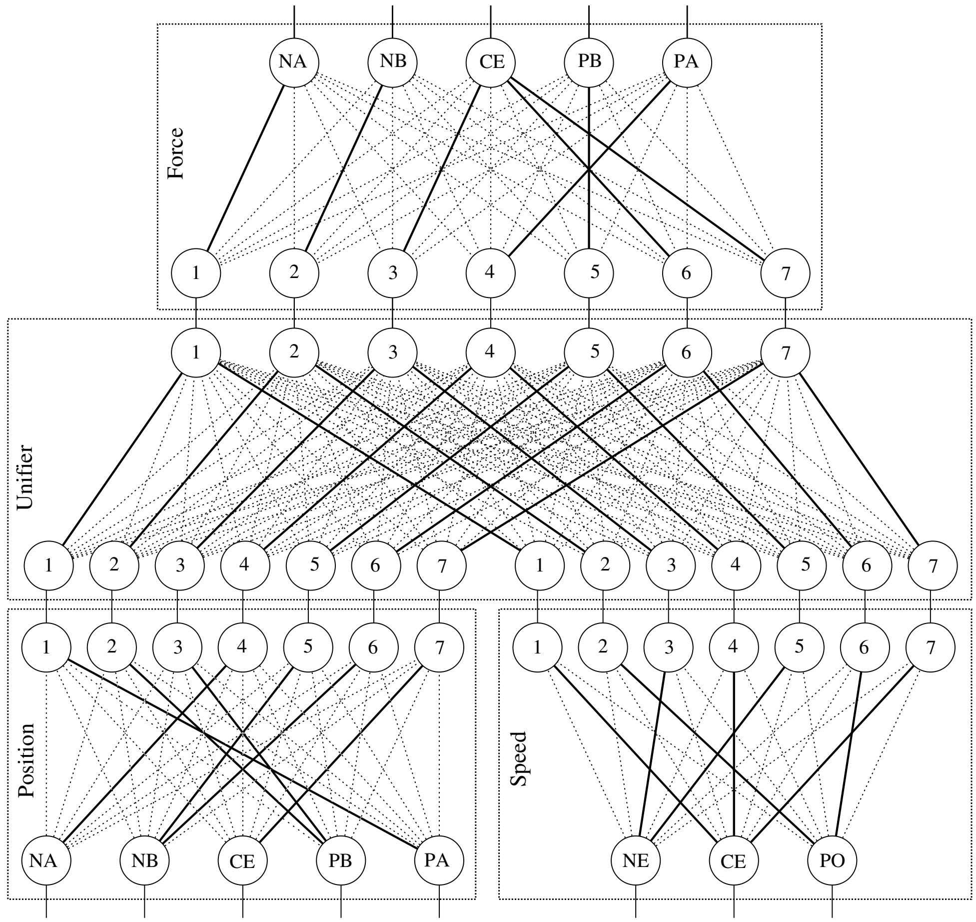

Figure 5. Set of CCLAMs that stores the rules of the braking system.

rules is the one shown in Figure 5 (in order to make the representation easier, the CCLAM associated with the force has been rotated so that its F_A layer is above and F_B is below). In the CCLAMs associated with the position, speed, and force, the connections with a weight equal to zero are represented using broken lines, and the connections with a weight equal to one are shown with solid lines. In the case of the unifier, the connections with a weight equal to one are represented using broken lines, whereas those with a weight equal to zero are shown with solid lines.

In order to determine the value of the force necessary to stop the moving object, we will use the t-norm of the minimum and the t-conorm of the maximum. Therefore, the function Υ of the CCLAMs associated with linguistic variables will be the minimum and the function Ψ will be the maximum.

Let us suppose that we want to stop a moving object for which the position and speed are:

- Position $\frac{1}{4}$ high negative, $\frac{3}{4}$ low negative
- Speed $\frac{1}{2}$ negative, $\frac{1}{2}$ zero

The patterns that code the situation are:

$$
\begin{aligned}
P &= \left(\tfrac{1}{4}, \tfrac{3}{4}, 0, 0, 0\right) \\
S &= \left(\tfrac{1}{2}, \tfrac{1}{2}, 0\right)
\end{aligned} \tag{13}
$$

As the pattern P is supplied to the CCLAM associated with the position, the signal propagates from F_A to F_B as shown in the following table. In the cell corresponding to row S and column T, the returned result by $\Upsilon^C(P_T, m^C_{T,S}) = \min(P_T, m^C_{T,S})$ appears. At the end of each row is the value provided by Ψ^C for the values of that row.

Υ^C	HN	LN	ZE	LP	HP	Ψ^C
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$
5	0	$\frac{3}{4}$	0	0	0	$\frac{3}{4}$
6	0	$\frac{3}{4}$	0	0	0	$\frac{3}{4}$
7	0	0	0	0	0	0

In the same way, since the pattern S is provided to the input of the CCLAM associated with the speed, the signal propagated from F_A towards F_B is shown in the following table. In the cell corresponding to row S and column T, the result of the function $\Upsilon^C(P_T, m^C_{T,S}) = \min(P_T, m^C_{T,S})$ appears. At the end of each row is the value returned by Ψ^C for the values of that row.

Υ^C	NE	ZE	PO	Ψ^C
1	0	$\frac{1}{2}$	0	$\frac{1}{2}$
2	0	0	0	0
3	$\frac{1}{2}$	0	0	$\frac{1}{2}$
4	0	$\frac{1}{2}$	0	$\frac{1}{2}$
5	$\frac{1}{2}$	0	0	$\frac{1}{2}$
6	0	0	0	0
7	0	$\frac{1}{2}$	0	$\frac{1}{2}$

The result obtained in the two previous CCLAMs is then sent to the unifier. The pattern $(0, 0, 0, \frac{1}{4}, \frac{3}{4}, \frac{3}{4}, 0)$ is received from the memory associated with position, and the pattern $(\frac{1}{2}, 0, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2})$ is received from the memory associated with speed.

The signal propagated from F_A towards F_B is shown in the following table. In the cell corresponding to row S and column T, the value appears $\Upsilon^C(P_T, m^C_{T,S}) =$

$\max(P_T, m^C_{T,S})$. At the end of each row is the value returned by $\Psi^C = \min$ for the values of that row.

Υ^C	POSITION							SPEED							Ψ^C
	1	2	3	4	5	6	7	1	2	3	4	5	6	7	
1	0	1	1	1	1	1	1	$\frac{1}{2}$	1	1	1	1	1	1	0
2	1	0	1	1	1	1	1	1	0	1	1	1	1	1	0
3	1	1	0	1	1	1	1	1	1	$\frac{1}{2}$	1	1	1	1	0
4	1	1	1	$\frac{1}{4}$	1	1	1	1	1	1	$\frac{1}{2}$	1	1	1	$\frac{1}{4}$
5	1	1	1	1	$\frac{3}{4}$	1	1	1	1	1	1	$\frac{1}{2}$	1	1	$\frac{1}{2}$
6	1	1	1	1	1	$\frac{3}{4}$	1	1	1	1	1	1	0	1	0
7	1	1	1	1	1	1	0	1	1	1	1	1	1	$\frac{1}{2}$	0

The result obtained by the unifier indicates that the applicable rules are the fourth and the fifth with degrees $\frac{1}{4}$ and $\frac{1}{2}$, respectively. The pattern obtained, $(0, 0, 0, \frac{1}{4}, \frac{1}{2}, 0, 0)$, is sent to the CCLAM associated with force.

The signal propagated from F_B towards F_A in this CCLAM is shown in the following table. In the cell corresponding to row S and column T, the result given by the function $\Upsilon^R(P_T, m^R_{T,S}) = \min(P_T, m^R_{T,S})$ appears. At the end of each row is the value returned by $\Psi^R = \max$ for the values of that row.

Υ^R	1	2	3	4	5	6	7	Ψ^R
HN	0	0	0	0	0	0	0	0
LN	0	0	0	0	0	0	0	0
ZE	0	0	0	0	0	0	0	0
LP	0	0	0	0	$\frac{1}{2}$	0	0	$\frac{1}{2}$
HP	0	0	0	$\frac{1}{4}$	0	0	0	$\frac{1}{4}$

Therefore, in order to stop a moving object that is at a position $\frac{1}{4}$ high negative, $\frac{3}{4}$ low negative, and which moves with a speed $\frac{1}{2}$ negative, $\frac{1}{2}$ zero, it is necessary to apply a force $\frac{1}{2}$ low positive, $\frac{1}{4}$ high positive to it.

5. CONCLUSIONS

- The continuous classifying associative memory (CCLAM) allows imprecise information expressed in linguistic terms to be correctly stored and recovered.
- The correct choice of the functions that define the memory behavior allows CCLAM groupings to be composed to fulfill the requirements of information storage and recovery.
- The storage of rules with CCLAM not only enables us to recover rules but also to obtain the appropriate consequent for any antecedent.

References

1. Bailón AB, Delgado M, Fajardo W. Clasificación de información con memorias asociativas. In: Actas de la VII Conferencia de la Asociación Española para la Inteligencia Artificial, 1997.

2. Bailón AB, Delgado M, Fajardo W. Memoria asociativa clasificadora mejorada. In: Actas de la VIII Conferencia de la Asociación Española para la Inteligencia Artificial, 1999.
3. Bailón AB, Delgado M, Fajardo W. CLAM: A new model of associative memory. Int J Intell Syst 2000;15:549–564.
4. Bailón AB, Delgado M, Fajardo W. Continuous classifying associative memory. (Manuscript submitted to the Int J Intell Syst, 2001.)
5. Bailón AB. Memorias asociativas difusas: Diseño e implementación. PhD thesis, Universidad de Granada, 2001.
6. Kosko B. Adaptive inference. Technical report, Verac, 1985.
7. Kosko B. Fuzzy associative memories. In: Kandel A, editor. Fuzzy expert systems. Addison-Wesley; 1987.
8. Zadeh LA. A theory of commonsense knowledge. In: Termini TS, editor. Aspects of vagueness. Reidel; 1984. pp 257–295.