

# A Recursive Pattern Recognition Algorithm

*Puerto Cuadros Eduard Gilberto<sup>1</sup>, Aguilar Castro Jose Lisandro<sup>2</sup>*

<sup>1</sup>*Facultad de Ingeniería, Universidad Francisco de Paula Santander. Cúcuta, Colombia.  
eduardpuerto@ufps.edu.co*

<sup>2</sup>*Facultad de Ingeniería, Universidad de Los Andes. Mérida, Venezuela.  
Prometeo researcher, Universidad Técnica Particular de Loja, Ecuador.  
aguilar@ula.ve*

## Abstract

This work presents a pattern recognition algorithm based on the systematic functioning of the human neocortex, in processes of pattern recognition. The algorithm exploits the idea of recursivity, based on the neocortical hierarchy and the decomposition/integration in the recognition process. The proposed algorithm is tested to analyze its capabilities of pattern recognition.

**Keywords:** recursivity; pattern; pattern recognition modules.

## Un Algoritmo Recursivo de Reconocimiento de Patrones

### Resumen

En este trabajo, proponemos un algoritmo de reconocimiento de patrones basado en el funcionamiento sistemático del neocórtex humano, en procesos de reconocimiento de patrones. El algoritmo explota la idea de recursividad, basada en la jerarquía neocortical y en la desagregación/integración del patrón en el proceso de reconocimiento. El algoritmo propuesto es probado para analizar sus capacidades de reconocimiento de patrones.

**Palabras Clave:** recursividad; patrón; módulo de reconocimiento de patrones.

### 1. Introducción

En general, existen muchos algoritmos para la resolución del problema de reconocimiento de patrones desde hace tiempo [1]. Algunos de ellos, basados en modelos computacionales del cerebro, tal como la Teoría de Resonancia Adaptativa [2], o el modelo neuronal aleatorio [3]. Actualmente, el aprendizaje profundo o *Deep Learning*, ha evolucionado y perfeccionado el reconocimiento de patrones [4] [5]. Las arquitecturas bajo aprendizaje profundo generan modelos composicionales, donde el objeto (a analizar y reconocer) es expresado como una composición de niveles de imágenes primitivas (parecido al modelo presentado en la sección 2). Por otra parte, existen

otros modelos relacionados con el funcionamiento parcial del neocórtex, que revelan como el cerebro representa o mantiene la información en memoria [6] [7]. Además, se está desarrollando tecnología basadas en los principios que rigen el neocórtex [8], como en *Numenta*, compañía que basa toda su tecnología en la *Hierarchical Temporal Memory* (HTM), teoría computacional sobre el neocórtex [9].

Nuestra propuesta del algoritmo de reconocimiento de patrones recursivo adopta teorías sobre el funcionamiento del neocórtex del cerebro humano (la región del cerebro responsable de las funciones cognitivas de alto nivel) [10], [11]. La teoría de la mente basada en el reconocimiento de patrones (en adelante, PRTM por sus siglas en inglés), describe un modelo del funcionamiento del neocórtex

sobre la base que el proceso de reconocimiento de patrones en el neocórtex se realiza a través de módulos de reconocimiento de patrones, conformados por unas 100 neuronas, y que estos módulos (y los patrones) están organizados en niveles, reflejando así una organización jerárquica del mismo. Para el diseño del algoritmo de reconocimiento de patrones recursivo, introducimos la formalización lógico-matemática necesaria. El algoritmo es probado en un problema de reconocimiento de patrones, para determinar su capacidad de reconocimiento.

## 2. Modelo de la Mente Basado en el Reconocimiento de Patrones: Una Síntesis de PRTM.

En PRTM se definen los siguientes aspectos del funcionamiento del cerebro humano [11]: i) Nuestra memoria se maneja como una jerarquía de patrones, ii) Si sólo percibimos una parte de un patrón (mediante la vista, el oído, o el olfato), podemos reconocerlo.

Por otra parte, PRTM presupone varias hipótesis sobre la estructura del neocórtex biológico [11]: i) Uniformidad de la estructura básica del neocórtex, denominada columna cortical. Ellas son los *módulos de*

*reconocimiento de patrones neocortical* para PRTM; ii) Los módulos de reconocimiento de patrones se conectan entre sí durante todo el tiempo.

Pasemos a describir el módulo de reconocimiento de patrones de PRTM (ver figura 1): a) Cada una de las dendritas (cada círculo de color mostaza) envía información (parámetros de tamaño, importancia, variabilidad (f)) hacia el interior del módulo, indicando la presencia de un patrón en el nivel inferior, o hacia el exterior, indicando que se está a la espera del reconocimiento del respectivo patrón. Cuando hay reconocimiento, se genera una salida (b). Cada patrón se hace/define según un orden secuencial de las señales de entrada, para que el reconocedor de patrones sea capaz de reconocerlo. Por otra parte, si un reconocedor de patrones de nivel más alto recibe una señal procedente de casi todos los reconocedores que conforman su entrada, entonces este reconocedor es posible que envíe una señal excitadora hacia los reconocedores de más bajo nivel de los patrones faltantes (vía una dendrita), para indicar que está a la expectativa de los mismos (caso (c)). Además, existen señales inhibitorias procedentes tanto de un espacio de reconocimiento de más bajo nivel (d), como de más alto nivel (e), que pueden inhibir el proceso de reconocimiento de un patrón.

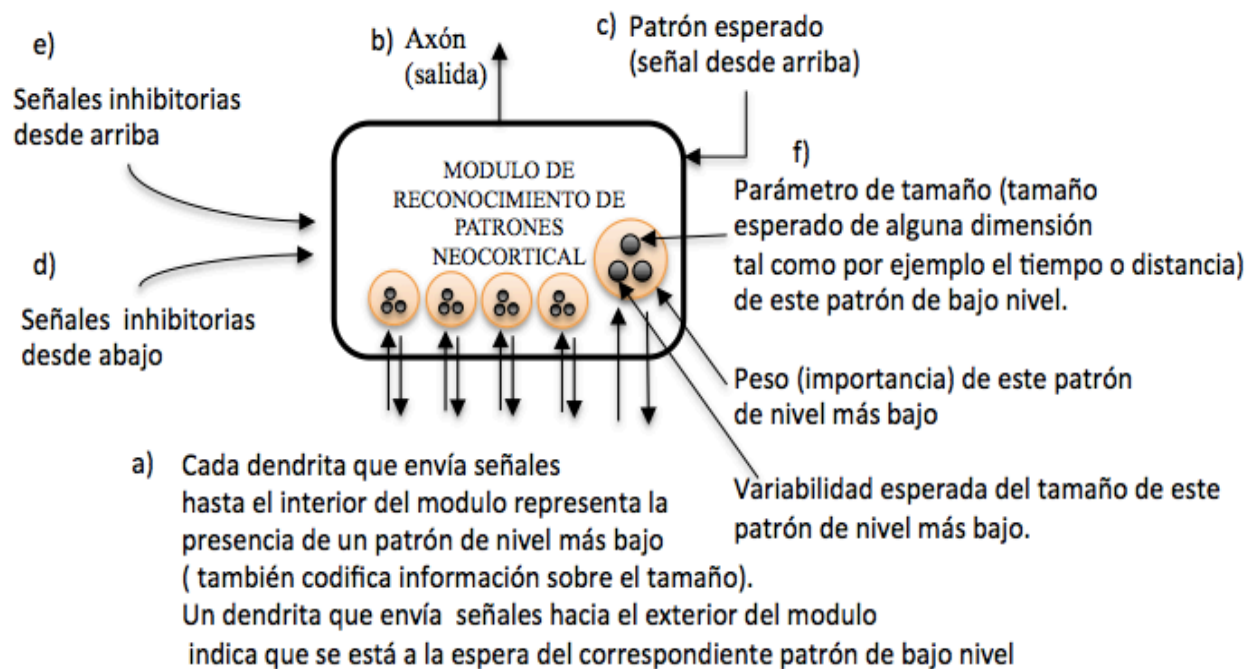


Figura 1. Módulo de reconocimiento de patrones neocortical PRTM [11].

### 3. Modelo Matemático del Algoritmo Recursivo de Reconocimiento de Patrones propuesto

El esquema general del proceso recursivo del modelo PRTM se muestra en la figura 2 en adelante denominado AR2P por sus siglas en español. El sistema general de reconocimiento recibe como entrada un patrón  $s()$ , que viene del sistema de percepción. Entonces, el sistema toma  $s()$  e inicia su proceso de reconocimiento. Para ello, dispone de varios niveles de reconocimiento, representados como óvalos en el esquema, e identificados como  $\chi_i$ , desde  $i=1$  hasta  $m$ .  $\chi_1$  es el nivel de reconocimiento para patrones atómicos, y  $\chi_m$  es nivel de reconocimiento para patrones complejos.

Cada nivel está compuesto por varios módulos de reconocimiento de patrones  $\Gamma_{ji}$  (para  $j=1,2,3, \dots, \#$  de módulos en el nivel  $i$ ), que conforman el espacio de reconocimiento del nivel  $\chi_i$ . Cada módulo tiene la función de reconocer su patrón respectivo  $\rho_{ji}$ . La función de cada reconocedor de patrones es la de reconocer su correspondiente patrón. Veamos con más detalle el modelo:

- Las función de entrada  $s()$ , representa la presencia de un patrón a reconoce
- Existe una relación  $v$  de composición jerárquica estructural entre los  $\Gamma_{ji}$  de los diferentes niveles  $\chi_i$ , tal que  $v: \Gamma_{rt} \rightarrow \Gamma_{lk}$ , donde  $t < k$ , y la relación " $\rightarrow$ " indica que los  $\Gamma_{rt} \in \chi_t$  están contenidos o hacen parte de  $\Gamma_{lk} \in \chi_k$  de más alto nivel. En otras palabras, un  $\Gamma$  de  $\chi_k$  se compone de diferentes  $\Gamma$  del  $\chi_t$  de más bajo nivel.
- Cada nivel  $i$  produce una señal de salida (de reconocimiento o aprendizaje) en función de las respuestas

de sus módulos.

**Prueba:** Sea  $\rho_{ji}$  un patrón reconocido por el módulo  $\Gamma_{ji}$  (módulo  $j$  de reconocimiento de patrones del nivel  $i$ ). Si  $\rho_{ji}$  tiene un número finito de sub-patrones que lo componen, entonces los módulos  $\Gamma$  que los reconocen no pueden encontrarse en el mismo nivel de reconocimiento de  $i$ , sino en un nivel más abajo ( $i-1$ ) en la jerarquía. Así, las salidas de los módulos de reconocimiento de más bajo nivel se convierten en señales de reconocimiento para los módulos de más alto nivel que los contienen, estableciendo de esta manera la relación  $v: \Gamma_{rt} \rightarrow \Gamma_{lk}$ .

#### 3.1 Estructura Formal del Módulo de Reconocimiento de Patrones AR2P

Un módulo  $\Gamma$  de reconocimiento de patrones para el algoritmo recursivo se define formalmente como una 3-tupla  $\Gamma = \langle E, U, S_o \rangle$  (ver tabla 1), donde  $E$ : es una matriz conformada por la 2-tupla  $E = \langle S, C \rangle$ , con  $S: S = \langle \text{Señal, Estado} \rangle$ , matriz que representa el conjunto de señales que conforman al patrón reconocido por  $\Gamma$  y sus respectivos estados. La variable estado es "true" cuando está presente la señal y "false" en caso contrario. Cada señal presente en  $\Gamma$  tiene su correspondiente reconocedor del patrón de nivel más bajo, y cuando ha sido reconocido su estado es "true". El número de señales que conforman al patrón es específico para cada módulo  $\Gamma$ .  $C = \langle D, V, P \rangle$  es la matriz que codifica información sobre el patrón:  $D$  son los descriptores propios del patrón en  $\Gamma$ ;  $V$  es el vector de dominio del descriptor o variabilidad esperada del mismo (por ejemplo, "de tiempo", "de distancia", etc.); y  $P$  es el peso de importancia del descriptor para el reconocimiento del patrón.

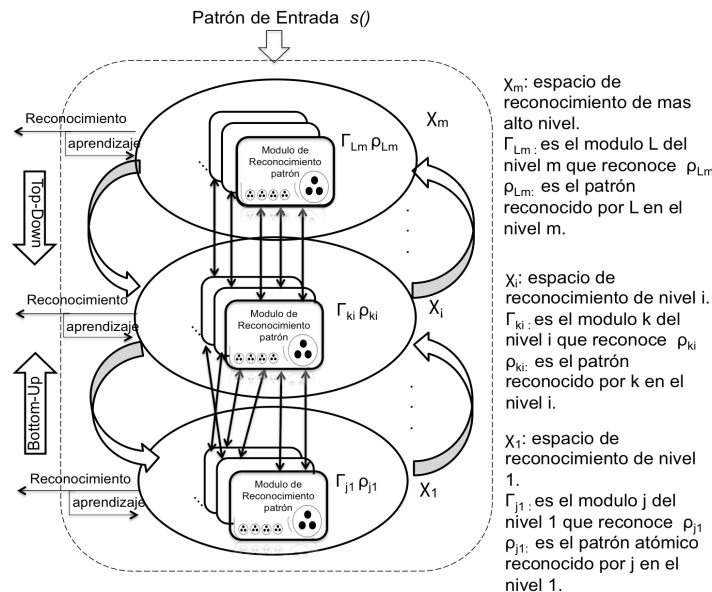


Figura 2. Sistema jerárquico de reconocimiento de patrones AR2P.

$\underline{U}$ : Es el vector de umbrales usado por el módulo  $\Gamma$  para el reconocimiento del patrón. Existen dos tipos de umbrales:  $\Delta U1$  es el umbral de reconocimiento usado por el algoritmo para el caso del reconocimiento del patrón por señales claves, y  $\Delta U2$  es el umbral de reconocimiento usado por el algoritmo para el caso del reconocimiento del patrón por mapeo parcial y/o total.

$\underline{So}$ : Señal de salida. Cada módulo  $\Gamma$  produce una única señal de salida ( $So$ ) de reconocimiento del patrón hacia los

niveles superiores, o señales de petición hacia niveles más bajos de reconocimiento. Cuando hay reconocimiento, esta señal  $So$  es la salida del sistema, y se difunde a niveles superiores alcanzables, haciendo que se modifique el estado de la señal a "true" en los patrones de más alto nivel del cual hace parte.  $So$  como petición se convierte en la señal de entrada  $s()$  para los módulos de reconocimiento de patrones de nivel inferior. La tabla 1 muestra la estructura de datos definida para Matriz E.

**Tabla 1**  
Matriz E = <S,C>.

E				
S		C		
Señal	Estado	Descriptor (D)	Dominio de Valores(V)	Peso (P)
1	False	Descriptor1	<Descriptor 1>	[0,1]
2	False	Descriptor2	< Descriptor 2>	[0,1]
3	False	Descriptor3	< Descriptor 3 >	[0,1]
...	...	...	...	...
n	False	Descriptorn	< Descriptor n>	[0,1]

< $\Delta U1, \Delta U2$ >

### 3.2 Teoremas para el Reconocimiento de un Patrón

Para reconocer un patrón se han definido dos *teoremas*:

**Teorema 1:** *Reconocimiento por señales claves.* Usa los descriptores con mayor peso de importancia para el patrón a reconocer y el umbral  $\Delta U1$ , para el reconocimiento. Este teorema establece que se reconoce un patrón  $\rho$  por señales claves, si la suma de los descriptores de mayor importancia reconocidos es mayor al umbral. Los descriptores de

mayor importancia son denominados señales claves para ese patrón.

**Definición señal clave.** Una señal  $S$  activa en el módulo  $\Gamma$  es clave si su peso de importancia tiene un valor mayor o igual al peso promedio de todas las señales en  $\Gamma$ . Sean  $S(\Gamma) = \{S_i \mid S_i \in \Gamma, \text{ con } i=1, \dots, n\}$ ,  $P(S_i)$  el peso de importancia de la señal particular  $S_i$ ,  $P_m(S(\Gamma))$  el peso promedio de las señales que hacen parte de  $\Gamma$ , y  $Clave_\Gamma = \{S_i \mid S_i \in \Gamma \wedge S_i \text{ es clave}\}$  el conjunto de señales claves, la ecuación siguiente establece quienes son las señales claves.

$$\forall S_i \in S(\Gamma), \text{ Si } [P(S_i) \geq P_m(S(\Gamma))] \rightarrow S_i \in Clave_\Gamma \tag{1}$$

Este teorema matemáticamente se formaliza de la siguiente manera:

$$\frac{\sum_{i=1}^n |s_i \in S(\Gamma) \cap Estado(s_i=true) \cap s_i \in Clave_\Gamma| \cdot S_i}{|Clave_\Gamma|} \geq \Delta U1 \rightarrow \text{reconoce patrón} \tag{2}$$

**Teorema 2:** Reconocimiento por mapeo() parcial. Usa para el reconocimiento a todas las señales. Este teorema establece que se reconoce un patrón  $\rho$

por *mapeo() parcial*, si la suma de los descriptores reconocidos es mayor al umbral  $\Delta U2$ . Este teorema se formaliza como sigue:

$$\frac{\sum_{i=1}^n | s_i \in S(\Gamma) \cap \text{Estado}(s_i = \text{true}) | \cdot s_i}{n} \geq \Delta U2 \rightarrow \text{reconoce patrón} \quad (3)$$

Estos teoremas son usados por cada módulo de cada nivel de reconocimiento  $X_i$  (desde  $X_2$  hasta  $X_m$ ) de AR2P, excepto por el nivel atómico.

#### 4. Modelo Computacional del Algoritmo Recursivo de Reconocimiento de Patrones AR2P Propuesto

módulo de nivel superior invoca a los módulos del nivel inferior que lo constituyen, y estos de manera recursiva hacen lo mismo (ver figura 2).

En el algoritmo de reconocimiento del patrón, el

*Algoritmo Recursivo de Reconocimiento de Patrones AR2P*

1.  $y = s()$
2.  $L = \text{descomponer}(y)$  // extraer sub-patrones
3. Determinar el nivel  $X_i$  de la jerarquía como inicio del reconocimiento de  $y$
4. Crear  $L$  solicitudes de petición de reconocimiento de  $(y)$
5. IF  $y = \text{señales de más bajo nivel } X_{i-1}$  o tiene señales atómicas reconocidas THEN // regreso de la recursión.
6. Calcular reconocimiento  $(y)$
7. IF calcular exitoso por cualquier estrategia THEN
8. Realizar Aprendizaje por refuerzo
9. IF Ultimo nivel de la Jerarquía  $X_m$  y  $X_m \neq X_{i-1}$  //  $X_m$  del  $s()$  inicial
10. Enviar  $So$  de reconocimiento  $(Y)$  salida del sistema.
11. ELSE
12. Crear salida  $So$  // armar la señal a difundir
13. Enviar la salida  $So$  de reconocimiento de  $(y)$  a nivel superior // hay difusión.
14. FIN IF
15. ELSE // caso cuando no se reconoce algo nuevo
16. IF señales atómicas reconocidas // Todas.
17. Realizar aprendizaje nuevo en ese nivel.
18. Crear  $So$  // armar la señal a difundir.
19. Enviar  $So$  de reconocimiento de  $(y)$  a nivel superior //difusión
20. ELSE ignorar //información insuficiente para aprender nuevo patrón
21. FIN IF
22. FIN IF // cierre de IF línea 7
23. FIN IF // cierre del IF de señales atómicas línea 5.
24. ELSE //
25. Enviar  $L$  solicitudes de petición de reconocimiento de  $(y)$  // a los niveles inferiores.
26. Recibir  $L$  respuestas de los niveles inferiores //
27. Calcular reconocimiento de  $(y)$
28. IF calcular exitoso por cualquier estrategia THEN
29. Realizar Aprendizaje por refuerzo
30. IF Ultimo nivel de la Jerarquía  $X_m$  //  $X_m$  del  $s()$  inicial
31. Enviar  $So$  de reconocimiento  $(Y)$  salida del sistema.
32. ELSE // seguir subiendo en el reconocimiento
33. Crear  $So$  // armar la señal a difundir.
34. Enviar  $So$  de reconocimiento de  $(y)$  a nivel superior //difusión
35. FIN IF

```

36.      ELSE // caso cuando no se reconoce algo nuevo
37.      IF señales atómicas reconocidas // Todas.
38.      Realizar aprendizaje nuevo en ese nivel.
39.      Crear  $S_o$  // armar la señal a difundir.
40.      Enviar  $S_o$  de reconocimiento de ( $y$ ) a nivel superior //difusión
41.      ELSE ignorar //información insuficiente para aprender nuevo patrón
42.      FIN IF
43.      FIN IF
44.      FIN IF

```

**Figura 3.** Algoritmo recursivo de reconocimiento de patrones AR2P.

El algoritmo funciona de la siguiente manera, el sistema de reconocimiento recibe un patrón de entrada a reconocer  $s()$  (línea 1) y genera señales de reconocimiento de  $s()$  como señal de salida  $S_o$  en diferentes casos (líneas 10, 13, 19, 31, 34, 40). Además, tiene dos pasos para aprender: uno para cuando hubo reconocimiento del patrón de entrada, en ese caso hay un aprendizaje por refuerzo (líneas 8, 29); y otro cuando no hubo reconocimiento ya que es un patrón nuevo (líneas 17, 38). Los dos mecanismos de aprendizaje son *Aprendizaje\_nuevo* y *Aprendizaje\_por\_refuerzo*. El primero crea un nuevo módulo de reconocimiento para un nuevo patrón y el segundo ajusta el módulo a los cambios existentes en el patrón de entrada. Para más detalle de estos procesos de adaptación ver [12], donde se especifican los algoritmos de aprendizaje propuestos.

Recibida  $s()$ , se descompone en sus sub-patrones (línea 2), y se determina el nivel  $X_i$  de la jerarquía como inicio del reconocimiento de  $s()$  (línea 3). Fijado el nivel, se crea  $L$  peticiones de reconocimiento de ( $y$ ) a través de  $s()$  (línea 4). Si son patrones atómicos (nivel  $X_i$ ) se realiza el proceso de reconocimiento de los patrones atómicos (líneas 5-23), de lo contrario se envían  $L$  solicitudes a los niveles inferiores (línea 25), y se espera sus reconocimientos (línea 26).

A continuación se calcula el reconocimiento del patrón actual (línea 27). Si el reconocimiento fue exitoso (Ecs. 2 y 3) (línea 28), y si es el último nivel de la jerarquía ( $X_m$ , nivel donde inicio el reconocimiento), se realiza un aprendizaje por refuerzo (línea 29) y se genera una señal de salida  $S_o$  del módulo, que se convierte en la señal de salida del sistema "patrón reconocido" (línea 31). Si no es el tope de la jerarquía, se crea y se envía la señal  $S_o$  de reconocimiento de ( $y$ ) a los niveles superiores (líneas 33-34). Si no se reconoce nada y ese módulo ha recibido varias señales atómicas (línea 37), se realiza aprendizaje nuevo (línea 38) y se reconoce como patrón nuevo (línea 39), por lo cual se envía la señal de salida  $S_o$  de reconocimiento por aprendizaje. En la línea 25 acontece la llamada recursiva del sistema. Las  $s()$  son recibidas por los módulos de

reconocimiento de más bajo nivel, que pueden seguir descomponiendo el patrón (línea 2) hasta llegar al nivel  $X_1$  (línea 5), donde se inicia el proceso de reconocimiento de patrones atómicos.

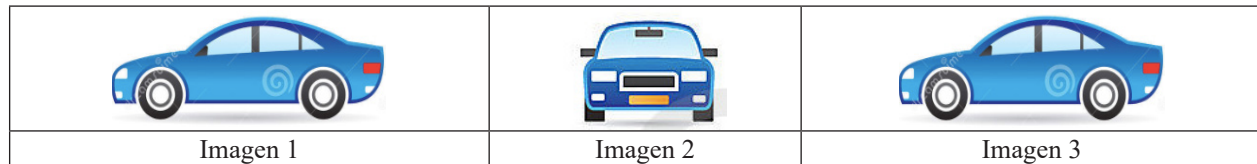
La descripción de la propuesta en términos recursivos, es la siguiente: el algoritmo reduce progresivamente la complejidad del patrón a reconocer (ver línea 25 del algoritmo de la figura 3), y al llegar al nivel atómico (patrones base no divisibles)  $X_1$  (ver línea 5 del algoritmo), inicia el proceso de reconocimiento de patrones, desde los atómicos hasta el más complejo. Todo ese proceso se resuelve de forma recursiva, tal que a medida que se descompone se invocan los respectivos módulo de reconocimiento de patrones del siguiente nivel, hasta llegar el nivel atómico, el cual regresa sí reconoció o no (líneas 6 al 23) a los módulos de reconocimiento de su nivel superior, los cuales a su vez realizan lo mismo (regreso de la llamada recursiva, ver líneas 27 al 41). El proceso de reconocimiento de los niveles superiores se basa en los teoremas 1 y 2. Así, vemos que el algoritmo se invoca a sí mismo (cuando realiza solicitudes de reconocimiento (línea 4 del algoritmo), y va regresando los resultados (si reconoció o no a su patrón) de cada llamada recursiva (líneas 7 al 23 del algoritmo). De esta manera, el caso base de la recursividad es el nivel atómico, y los procesos de descomposición y reconocimiento (basados en los teoremas 1 y 2) que se dan en los otros niveles son los casos generales. Además, entre ellos solo se envían señales indicando si se reconoció o no el patrón respectivo. Es decir, los parámetros que se envían es simplemente una señal, diciendo sí reconoció o no.

## 5. Parte Experimental

Para analizar la utilidad del algoritmo AR2P con respecto a otros métodos de reconocimiento, se desarrolla un caso de prueba para reconocer un *automóvil*, que permiten verificar la capacidad de manejar cualquier tipo de patrón, además, de cualquier tipo de complejidad, siguiendo un mismo procedimiento algorítmico.

Descripción del escenario. Un vehículo tiene múltiples vistas. Estas vistas dependen tanto del observador como del vehículo mismo. Por ejemplo, el ángulo de observación al

vehículo, el estado del vehículo (chocado, deteriorado, etc.), entre otros. La figura 4 muestra tres perfiles de observación de un automóvil: vista delantera, y dos vistas laterales.



**Figura 4.** Tres imágenes de perfil de un vehículo tipo automóvil.

Para el reconocimiento del patrón de la figura 4 (en cualquiera de sus tres vistas 1,2, o 3) usando AR2P, se consideran los siguientes sub-patrones: puertas, faros, ruedas, espejos, placas, etc.

Utilización del Algoritmo AR2P. Se asume que con la anterior información del escenario, se instancia AR2P para el reconocimiento del automóvil. Se definen dos espacios de reconocimiento:  $X_i=1$  para patrones atómicos, que en este escenario son los sub-patrones de análisis, i.e., las partes del vehículo; placa, espejos, ruedas, etc. El otro espacio es  $X_i=2$ , que en este escenario consta de tres módulos de reconocimiento, uno para cada perfil del vehículo (ver figura 4).

Para mostrar el algoritmo AR2P, se toma como entrada la imagen 2 de la figura 4. Sea  $y=s()$  la imagen 2 de la figura 4 (línea 1 del algoritmo), el siguiente paso consiste en descomponer la imagen del vehículo en los sub-patrones de análisis (línea 2). En este caso se tienen como sub-patrones: espejos retrovisores laterales, espejo retrovisor central, farolas delanteras, llantas delanteras y placa, etc. El proceso de descomposición particular se logra con métodos de segmentación de imágenes (por ejemplo, por métodos de detección de bordes). Una vez se tienen las partes de la imagen del vehículo particionadas, se determina el espacio de reconocimiento  $X_i$  en la arquitectura propuesta (ver figura 2), a partir del cual continuar con el proceso de reconocimiento. En esta prueba, el nivel de inicio de reconocimiento es el 2,  $X_i=2$ . Después, se procede a crear L solicitudes de petición de reconocimiento de las partes que conforman al vehículo o patrón  $y$  (línea 4). Así, dado que el nivel actual del proceso es  $X_i=2$ , para profundizar el proceso de análisis del patrón se envían L solicitudes de petición de reconocimiento de cada una de las partes del vehículo ( $y$ ) a los niveles inferiores (línea 25). Esta línea invoca el proceso recursivo, generándose nuevos procesos de reconocimiento en paralelo e independientes, para cada uno de los sub-patrones. De esta manera, el *proceso\_principal* invoca la recursividad. A continuación, se inician los nuevos procesos de reconocimiento invocados en la línea 25 por parte del *proceso\_principal*, para el respectivo

reconocimiento de las partes del vehículo contenidas en el patrón original, por ejemplo:  $L_1$ -sub-patrón de imagen espejo retrovisor lateral izquierdo, etc.

Veamos el caso del reconocimiento de  $L_1$ . Ahora  $y=s()$  es la sub-imagen de espejo retrovisor lateral izquierdo de la imagen 2 de la figura 4. El siguiente paso (línea 2), consiste en descomponer la sub-imagen del vehículo en los sub-patrones de análisis, en este caso el conjunto de sub-patrones de análisis es vacío. Por cuanto es un patrón atómico, se determina que el espacio de reconocimiento  $X_i$  es el nivel 1,  $X_i=1$ . Ya estando en el nivel  $X_i=1$  de inicio de reconocimiento, se procede a crear L solicitudes de petición de reconocimiento de las partes que conforman al vehículo o patrón  $y$  (línea 4). L contiene sólo el patrón de entrada por ser indivisible. Dado que el nivel actual del proceso es  $X_i=1$  se cumple la condición para realizar el cuerpo de instrucciones entre las líneas 6 y 23. Se procede entonces a realizar el reconocimiento del patrón del espejo retrovisor lateral ( $y$ ) (línea 6), y consiste en mapear dicho patrón (bajo alguna dimensión) con la base de datos de patrones atómicos que constituye el espacio de reconocimiento  $X_i=1$ . Damos por hecho que el mapeo es exitoso (línea 7), y se procede a aprender (ver [12] para más detalles). Se mira si el proceso está en el último nivel de la jerarquía  $x_m$ . Como no se ha alcanzado el último nivel (que es  $x_m=2$ ), se pasa al ELSE del condicional (línea 11), que consiste en crear la salida  $So$ , y enviar dicha salida  $So$  de reconocimiento del patrón espejo retrovisor lateral al nivel superior (difusión de una señal de reconocimiento positivo (True) al *proceso\_principal*). En este punto se termina este proceso interno, que fue invocado en la Línea 25, donde quedo el proceso principal.

Este mismo proceso se aplica para las demás solicitudes de reconocimiento. Si asumimos que hubo reconocimiento de todas las partes, el proceso principal retoma el control del programa donde se hizo la solicitud de los L reconocimientos (línea 25), y se procede a instanciar la estructura de datos de la Tabla 1, con las respuestas recibidas de las solicitudes de reconocimiento (ver tabla 2).

La tabla 2 muestra las 9 señales que corresponden a los sub-patrones del proceso de descomposición de la imagen del vehículo, con todos los estados de los patrones reconocidos (estado=true). En este experimento se ha asignado el peso de importancia de acuerdo a su influencia para el reconocimiento. Por ejemplo, los descriptores de las señales 1, 2, 3, 5 y 8 tienen poco peso (0.5) puesto que son partes opcionales o accesorios. En cambio, las señales 4, 6, 7 y 9 que corresponden a los faroles, persiana, llantas y vidrios parabrisas, tienen bastante peso (1) pues son claves para el reconocimiento del vehículo. Continuado con el proceso de reconocimiento del proceso principal, en la línea 26 se reciben las 9 respuestas de los niveles inferiores de la solicitud L. Una vez recibidas

estas señales en el módulo de reconocimiento de  $X_2$  desde donde se invocó la solicitud de reconocimiento, se procede a calcular el reconocimiento de los patrones de los módulos que las contienen (línea 27). En este caso, el módulo de interés que vamos a analizar es el que entra en reconocimiento del perfil del vehículo frontal, imagen 2 de la figura 4. Para calcular el reconocimiento, se usan los teoremas 1 y 2. El teorema 1 es para el Reconocimiento del vehículo por señales claves. En este caso se ha definido el umbral  $\Delta U1 = 0.75$ . Aplicando la Ec(1), las señales claves son:  $Clave\Gamma = \{i=4,6,7,9\}$ . Aplicando la Ec(2) para verificar el teorema 1, se obtiene que  $1 > 0.75$ . Por tanto, hay reconocimiento por señales claves. Así, no se aplica el teorema 2.

**Tabla 2**  
Matriz E = Vehículo vista frontal

E				
S		C		
Señal	Estado	Descriptor (D)	Dominio (V)	Peso (P)
1	True	Imagen espejo retrovisor lateral izq.	<null>	0.5
2	True	Imagen espejo retrovisor lateral Der.	<null>	0.5
3	True	Imagen espejo retrovisor central	< null >	0.5
4	True	Imagen de farola	< null >	1
5	True	Imagen exploradora	< null >	0.5
6	True	Imagen persiana	< null >	1
7	True	Imagen llanta	< null >	1
8	True	Imagen placa	< null >	0.5
9	True	Imagen parabrisas delantero	< null >	1
<ΔU1, ΔU2>				

Después de eso, el proceso principal pasa a chequear el resultado del cálculo (línea 29), y si el cálculo es exitoso por cualquier teorema, se procede a ejecutar el aprendizaje (línea 29, ver [12] para detalles de aprendizaje). Realizado el proceso de adaptación, se chequea si el proceso de reconocimiento del patrón vehículo está en el último nivel de la jerarquía (Línea 30). En efecto, actualmente el proceso de reconocimiento está en  $X_m=2$ , que es el último nivel de la jerarquía, y coincide con el nivel de reconocimiento inicial. Por tanto, se crea la señal de salida

So de reconocimiento ( $y=s()$  inicial), correspondiente a la imagen 2 de la figura 4 (Línea 31).

## 6. Discusión de Resultados

El modelo propuesto formaliza computacionalmente el modelo teórico del funcionamiento biológico del neocórtex en tareas cognitivas de reconocimiento de patrones. Basado en dicho modelo teórico, el algoritmo de reconocimiento de patrones debe ser altamente recursivo y uniforme. La



uniformidad permite que el proceso recursivo se de en todos los niveles de la misma forma, sin importar la complejidad. Así, el algoritmo reconoce un patrón de entrada sin importar su nivel de complejidad ni la naturaleza del mismo (imagen, texto, sonido, etc.).

El algoritmo reconoce los patrones de entrada por auto-asociación jerárquica de patrones. El sistema usa dos estrategias para el reconocimiento (teoremas de reconocimiento), una por señales claves y otra por mapeo total o parcial. La primera explota las señales importantes que facilitan el reconocimiento, la segunda estrategia explota de manera total las señales de entrada. Ambas controlan su disparo de reconocimiento a través de umbrales particulares.

La importancia del algoritmo AR2P frente a otros métodos de reconocimiento de patrones, es que se basa en el modelo biológico del neocórtex cerebral en tareas de reconocimiento, descrito en [11], el cual es altamente recursivo y uniforme. El modelo recursivo permite descomponer el problema de reconocimiento de patrones en patrones más simples, lo que posibilita analizar patrones muy complejos. Los cálculos definidos en los teoremas 1 y 2 son muy simples, y por el enfoque distribuido, es muy fácil realizar una paralelización por nivel. Así, el costo computacional puede ser mejorado con respecto a otros enfoques, además, con un uso más eficiente de la memoria, al contemplar en una sólo estructura de datos abstracta (ver tabla 1) muchas posibles instancias de varios patrones de objetos que compartan subpatrones. Otros métodos son menos eficiente porque la descomposición del proceso de reconocimiento no es posible, y consumen específicos espacios de memoria por cada objeto almacenado. En específico, si el patrón a reconocer es de tamaño u orden  $N$ , este enfoque se basa en el reconocimiento de sus subpatrones de un tamaño menor que  $N$ . De esta forma, al ir reduciendo progresivamente la complejidad del reconocimiento (descomponiéndolo y determinando si es posible reconocerlo usando los teoremas 1 y 2, caso general de la recursividad), llegará un momento en que se reconoce simplemente al patrón atómico (calculado trivial). El patrón atómico es el caso base de la recursividad.

## Conclusiones

Este trabajo presenta un algoritmo de reconocimiento de patrones basado en la teoría de la mente propuesta por Ray Kruzweil. El algoritmo de reconocimiento de patrones usa dos estrategias de análisis: una basada en señales claves y otra en señales activas. El algoritmo se basa en una formalización matemática del concepto de patrón, así como en definiciones lógico-matemáticas de los mecanismos de reconocimiento. Por otro lado, el algoritmo opera sobre una jerarquía, lo cual es susceptible

de extenderse hacia un tratamiento simultáneo de los niveles (paralelismo).

El algoritmo de reconocimiento de patrones usa dos métodos de aprendizaje definidos, basado en un esquema de aprendizaje por reforzamiento, mejorando así su capacidad de reconocimiento y ampliando sus bases de conocimiento, sin perder su capacidad de reconocimiento. Los algoritmos propuestos actualmente se están extendiendo, para introducir la idea de aprendizaje profundo, de tal manera de poder descubrir automáticamente los descriptores característicos de los patrones.

## Agradecimientos

Dr. Aguilar ha sido parcialmente financiado por el Proyecto Prometeo del Ministerio de Educación Superior, Ciencia, Tecnología e Innovación de la República del Ecuador.

## Referencias Bibliográficas

- [1] Aguilar, J.; Hernández, L.: "Diseño e Implementación de un Sistema de Reconocimiento de Patrones para Análisis de Líquidos Biológicos", IEEE Latinoamerica Transactions, Vol. 7, No. 1(2009) 12-26.
- [2] Altamiranda, J., Aguilar, J. Hernández L.: "Sistema de reconocimiento de patrones de sustancias químicas cerebrales basado en minería de datos." Computación y Sistemas Vol. 19, No 1(2015), 89-107.
- [3] Aguilar, José: "A Color Pattern Recognition Problem Based on the Multiple Classes Random Neural Network Model", Neurocomputing (special issue: Hybrid Neurocomputing), Elsevier, 61, 71-83, 2004.
- [4] Deng, L., & Yu, D.: " Deep learning: Methods and applications". Foundations and Trends in Signal Processing, Vol. 7 (2014), 197-387.
- [5] Schmidhuber, J.: "Deep learning in neural networks: An overview". Neural Networks, 61, 85-117. 2015.
- [6] Postle, Bradley R.: "How Does the Brain Keep Information "in Mind"?" Current Directions in Psychological Science, Vol. 25, No. 3(2016) 151-156.
- [7] Miller, Kenneth D.: "Canonical computations of cerebral cortex Current opinion in neurobiology", Vol. 37 (2016) 75-84.
- [8] Numenta, «NuPIC. Numenta Platform for Intelligent Computing» 2016. [En línea]. Available: <http://numenta.com/>. [consultada Diciembre 2016].

- [9] C. Surpur, «Cortical Learning Algorithm as Implemented in NuPIC» 2014. [En línea]. Disponible: <http://chetansurpur.com/slides/2014/5/4/cla-in-nupic.html#1>. [consultada Diciembre 2016].
- [10] Hawkins J. and Blakeslee S.: "On intelligence", Macmillan, 2007.
- [11] Kurzweil R.: "How to create a mind", Penguin Books, 2013.
- [12] Puerto, E., Aguilar J.: "Learning Algorithm for the Recursive Pattern Recognition Model." Applied Artificial Intelligence, Vol. 30, No 7 (2016) 662-678.

Recibido el 18 de Enero de 2016  
En forma revisada el 24 de Abril de 2017