

Guía de uso para el
Neural Controller Version 1.0

Cuando un sistema es altamente no lineal, resulta complejo modelar todas las incertidumbres asociadas al proceso. En esta situación una alternativa de solución es emplear técnicas no convencionales de control como son redes neuronales y lógica difusa, que consideran el proceso como un todo. Esto puede ser visto simplemente como una caja negra a la cual se le “enseñan” los patrones de entrada y se le indica cuales son las correspondientes salidas.

Lo anterior sugiere que se deben contemplar dos etapas, una de entrenamiento y una de prueba. La etapa de entrenamiento es como se mencionó arriba y en la etapa de prueba el controlador recibe entradas diferentes a las que se le “enseñaron”, por lo que debe ser capaz de inferir la salida apropiada. En esta guía se presenta de manera cualitativa el funcionamiento del software con el propósito de proporcionar una herramienta de solución a problemas no lineales de control.

El software esta basado en la teoría ART de redes neuronales y de operadores fuzzy, en particular de la red neuronal FuzzyARTMAP. Detalles pueden consultarse en un anexo al final de la presente guía de operación.

PASO 1

Descomprimir el archivo Neural Controller.ZIP a su disco duro, esto generara un folder Neural Controller el cual contiene los siguientes archivos:

1. NeuralController.exe (ejecutable)
2. TrainFile.dat (Contiene los patrones de entrenamiento)
3. TestFile.dat (Contiene los patrones de prueba)

Se recomienda visualizar los archivos para comprender su contenido. El sistema Windows preguntará sobre el software apropiado para visualizar el archivo, realmente este archivo esta en formato texto por lo que cualquier procesador de texto puede abrir el archivo por ejemplo, WordPad.

El contenido del archivo TrainFile.dat se muestra en la Figura 1. Cada línea de este archivo contiene el correspondiente patrón de entrada y su salida asociada. Es decir se presenta el mapeo o relación entre entradas y salidas. Los vectores de entrada para este ejemplo en particular constan de 6 elementos y su correspondiente vector de salida, de 4 elementos. Este archivo normalmente es generado por el usuario con datos reales para el vector de entrada y codificados para el vector de salida en formato binario por ejemplo.

El contenido del archivo TestFile.dat se muestra en la Figura 2. En este caso se puede observar que únicamente se presentan los datos de entrada, dado que lo se desea conocer es el resultado, salida, predicción ó inferencia del controlador para cada uno de los patrones de entrada.

TrainFile - WordPad

Archivo Edición Ver Insertar Formato Ayuda

0.496810507	0.496435272	0.493433396	0.483208255	0.515853659	0.499718574	1.0	0.0	0.0	0.0
0.480675422	0.478424015	0.4684803	0.392120075	0.594277674	0.498592871	1.0	0.0	0.0	0.0
0.460225141	0.467636023	0.445684803	0.336772983	0.696810507	0.495684803	1.0	0.0	0.0	0.0
0.477298311	0.493621013	0.477579737	0.46641651	0.614915572	0.498592871	1.0	0.0	0.0	0.0
0.468574109	0.468949343	0.453095685	0.342964353	0.653001876	0.498874296	1.0	0.0	0.0	0.0
0.478236398	0.497842402	0.47804878	0.487429644	0.610318949	0.499437148	1.0	0.0	0.0	0.0
0.466041276	0.507317073	0.464727955	0.532551595	0.672514071	0.498874296	1.0	0.0	0.0	0.0
0.43358349	0.510412758	0.43011257	0.54108818	0.836210131	0.498968105	1.0	0.0	0.0	1.0
0.444277674	0.531238274	0.436585366	0.645403377	0.789212008	0.501219512	1.0	0.0	0.0	1.0
0.401969981	0.5315197	0.398405253	0.642776735	1	0.497091932	1.0	0.0	0.0	1.0
0.486585366	0.510694184	0.482551595	0.553095685	0.569981238	0.500187617	1.0	0.0	0.0	0.0
0.449437148	0.529174484	0.494746717	0.650093809	0.780487805	0.499530957	1.0	0.0	0.0	1.0
0.482551595	0.520356473	0.499530957	0.609474672	0.599906191	0.500187617	1.0	0.0	1.0	0.0
0.460225141	0.522795497	0.488461538	0.615103189	0.718761726	0.507035647	1.0	0.0	0.0	1.0
0.495497186	0.515103189	0.498874296	0.583020638	0.527861163	0.500750469	1.0	0.0	1.0	0.0
0.495403377	0.515384615	0.500469043	0.585553471	0.529080675	0.500938086	1.0	0.0	1.0	0.0
0.495778612	0.509099437	0.501031895	0.551782364	0.525515947	0.500562852	1.0	0.0	1.0	0.0
0.495121951	0.515290807	0.50065666	0.58424015	0.530487805	0.500938086	1.0	0.0	1.0	0.0
0.49521576	0.512288931	0.500938086	0.568292683	0.528986867	0.500938086	1.0	0.0	1.0	0.0
0.496247655	0.50891182	0.501031895	0.550093809	0.522889306	0.500844278	1.0	0.0	1.0	0.0
0.495590994	0.523827392	0.500938086	0.627110694	0.528986867	0.501313321	1.0	0.0	1.0	0.0
0.496435272	0.518949343	0.501594747	0.601500938	0.523639775	0.501031895	1.0	0.0	1.0	0.0
0.496341463	0.518011257	0.50065666	0.595590994	0.523639775	0.501407129	1.0	0.0	1.0	0.0
0.49727955	0.514258912	0.501125704	0.576454034	0.517917448	0.501031895	1.0	0.0	1.0	0.0
0.497467167	0.515947467	0.501125704	0.584709193	0.517073171	0.501125704	1.0	0.0	1.0	0.0
0.49793621	0.521669794	0.49934334	0.612664165	0.515666041	0.501500938	1.0	0.0	1.0	0.0
0.498499062	0.514446529	0.50206379	0.576829268	0.51163227	0.501125704	1.0	0.0	1.0	0.0
0.499061914	0.51988743	0.501969981	0.603846154	0.509380863	0.501219512	1.0	0.0	1.0	0.0
0.498874296	0.516510319	0.501594747	0.586679174	0.510131332	0.501219512	1.0	0.0	1.0	0.0
0.498874296	0.514634146	0.50272045	0.577298311	0.509756098	0.501219512	1.0	0.0	1.0	0.0
0.493527205	0.519606004	0.500750469	0.598874296	0.536397749	0.501125704	1.0	0.0	1.0	0.0

Inicio Explorando - Ciateq... Microsoft Word - D... TrainFile - Wor... 09:05 a.m.

Figura 1 Contenido de archivo TrainFile.dat

TestFile - WordPad

Archivo Edición Ver Insertar Formato Ayuda

0.496341463	0.518011257	0.50065666	0.595590994	0.523639775	0.501407129				
0.49727955	0.514258912	0.501125704	0.576454034	0.517917448	0.501031895				
0.497467167	0.515947467	0.501125704	0.584709193	0.517073171	0.501125704				
0.49793621	0.521669794	0.49934334	0.612664165	0.515666041	0.501500938				
0.498499062	0.514446529	0.50206379	0.576829268	0.51163227	0.501125704				
0.499061914	0.51988743	0.501969981	0.603846154	0.509380863	0.501219512				
0.498874296	0.516510319	0.501594747	0.586679174	0.510131332	0.501219512				
0.498874296	0.514634146	0.50272045	0.577298311	0.509756098	0.501219512				
0.493527205	0.519606004	0.500750469	0.598874296	0.536397749	0.501125704				
0.493621013	0.52739212	0.5	0.636022514	0.537335835	0.501031895				
0.512570356	0.518105066	0.501219512	0.594559099	0.442120075	0.501782364				
0.50206379	0.502157598	0.501125704	0.514540338	0.491463415	0.501031895				
0.700000000	0.200000000	0.200000000	0.700000000	0.200000000	0.200000000				
0.200000000	0.700000000	0.700000000	0.200000000	0.700000000	0.700000000				
0.200000000	0.700000000	0.700000000	0.200000000	0.200000000	0.200000000				

Inicio Explorando - Ciateq... Microsoft Word - D... TestFile - Word... 09:05 a.m.

Figura 2 Contenido de archivo TestFile.dat PASO 2

Ejecutar el archivo NeuralController.exe.

Solamente se emplearan dos botones de la barra de herramientas, el botón de **Train** y el botón de **Test**.

Al oprimir el botón Train aparecerá el dialogo que se muestra en la Figura 3. Con el botón "Browse" en la ventana de dialogo se puede buscar el archivo de entrenamiento, aquí por simplicidad se inicializa la ventana con el nombre del archivo TrainFile.dat. Una vez ubicado el archivo correcto se procede a oprimir OK. Aparecerá enseguida una ventana pequeña señalizando que el entrenamiento ha finalizado. Existen otros parámetros propios de la red que son modificables, ello se explicará mas adelante en la sección de **Consideraciones**.

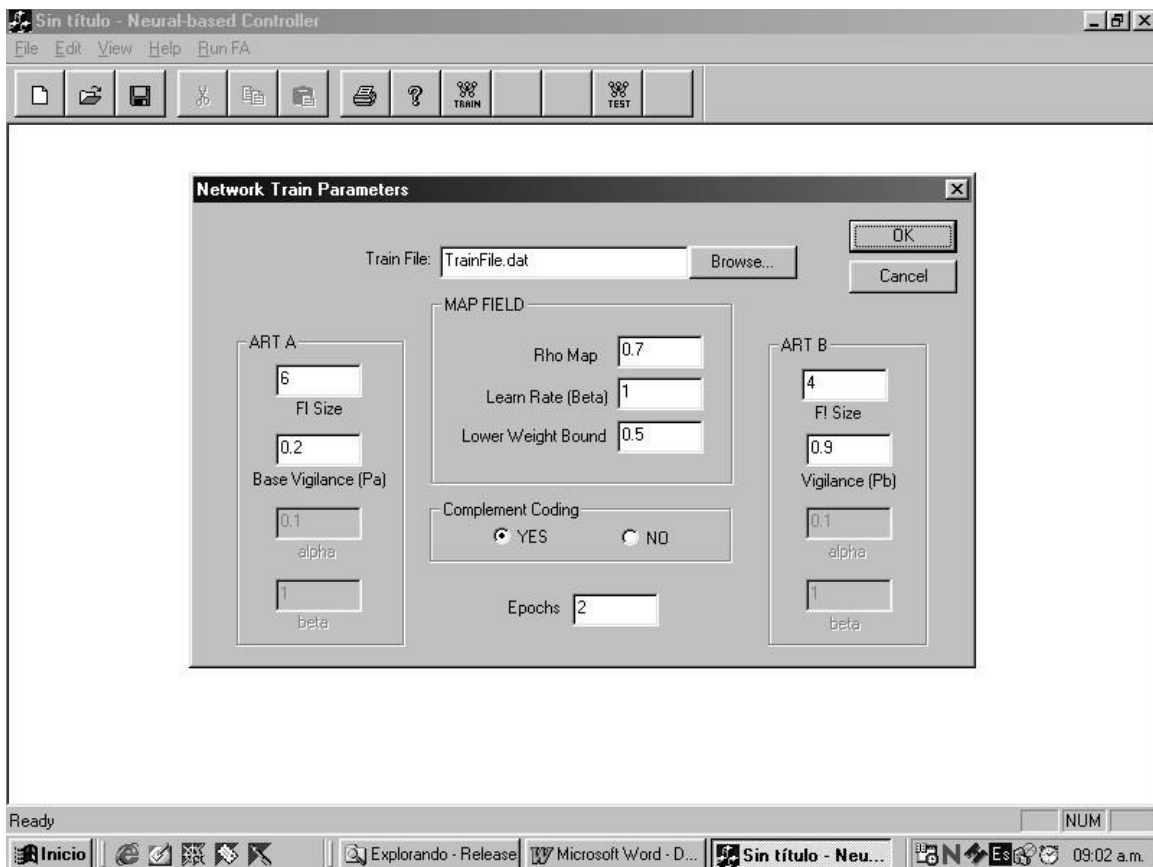


Figura 3 Ventana de Entrenamiento

PASO 3

Oprimir el botón Test.

Al oprimir el botón Test aparecerá el dialogo que se muestra en la Figura 4. De manera similar como en el caso anterior se ubicara el archivo de prueba (en el ejemplo, TestFile.dat) y se oprime OK, posteriormente el sistema también indicara cuando se concluya el proceso de prueba.

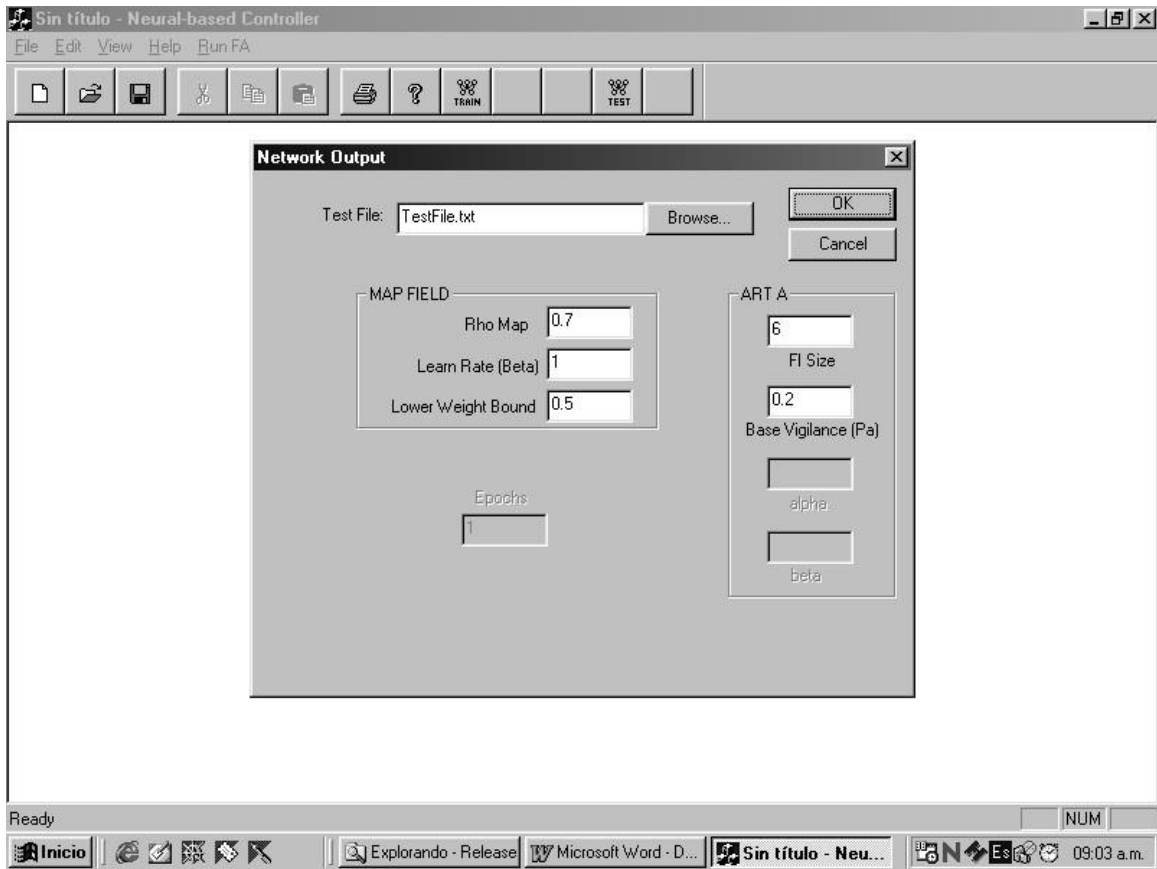
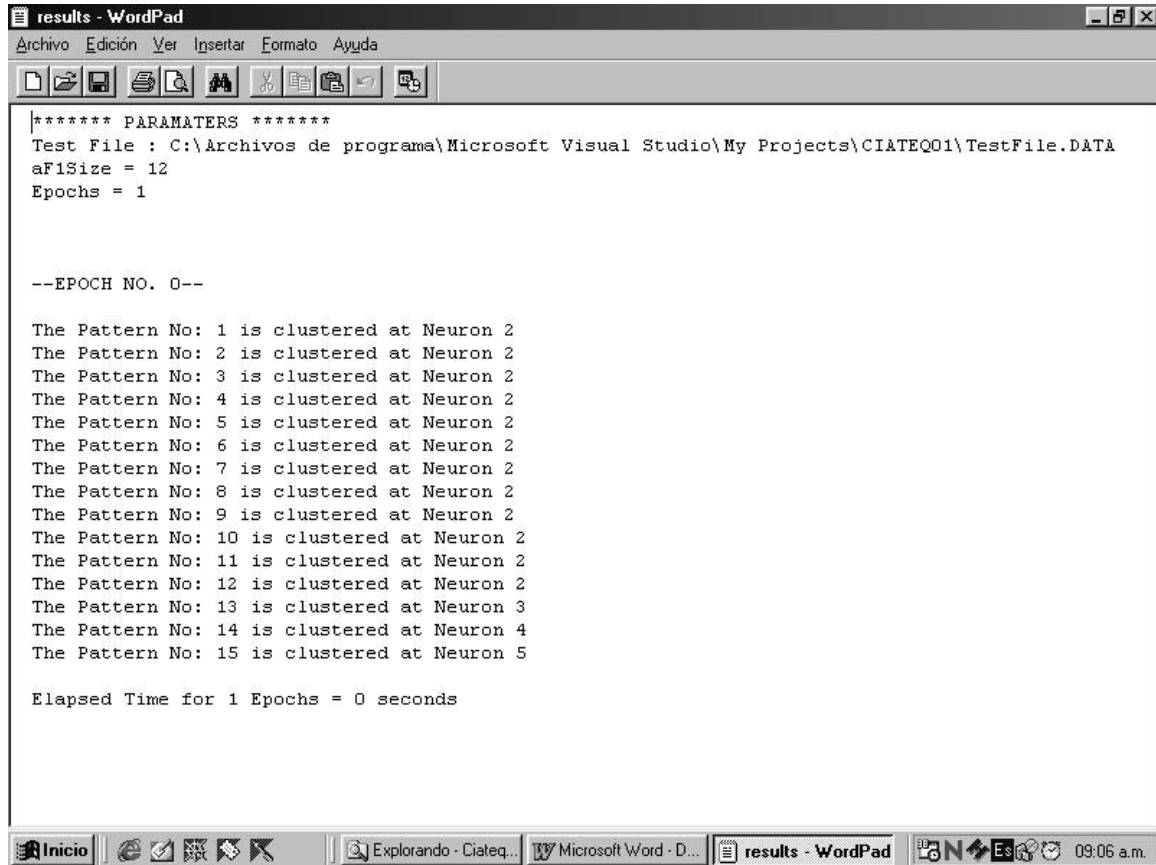


Figura 4 Ventana de Prueba

Los resultados pueden observarse en el archivo Results.dat como se muestra en la Figura 5.



```
***** PARAMATERS *****
Test File : C:\Archivos de programa\Microsoft Visual Studio\My Projects\CIATEQ01\TestFile.DAT
aFSize = 12
Epochs = 1

--EPOCH NO. 0--

The Pattern No: 1 is clustered at Neuron 2
The Pattern No: 2 is clustered at Neuron 2
The Pattern No: 3 is clustered at Neuron 2
The Pattern No: 4 is clustered at Neuron 2
The Pattern No: 5 is clustered at Neuron 2
The Pattern No: 6 is clustered at Neuron 2
The Pattern No: 7 is clustered at Neuron 2
The Pattern No: 8 is clustered at Neuron 2
The Pattern No: 9 is clustered at Neuron 2
The Pattern No: 10 is clustered at Neuron 2
The Pattern No: 11 is clustered at Neuron 2
The Pattern No: 12 is clustered at Neuron 2
The Pattern No: 13 is clustered at Neuron 3
The Pattern No: 14 is clustered at Neuron 4
The Pattern No: 15 is clustered at Neuron 5

Elapsed Time for 1 Epochs = 0 seconds
```

Figura 5. Resultados

En este archivo se encuentra información del archivo empleado, el tamaño del vector de entrada (multiplicado por 2 si se utilizó complement coding) y los resultados en cada época o presentación de patrones de entrada, para nuestro ejemplo únicamente se utilizó una época.

CONSIDERACIONES

Tanto los patrones de entrada como de salida deben pertenecer al intervalo (0,1). Si en cierta aplicación son mayores, estos deberán ser normalizados.

Los parámetros que pueden ser editados tanto en la etapa de entrenamiento como de prueba son:

Aquellos asociados al parámetro de vigilancia:

Base Vigilance (ρ_a)

Vigilance (ρ_b)

Rho map (ρ_{map})

Asociados al tamaño de los vectores de entrada y salida:
F1 Size

La opción de código de complemento:
Complement coding

Y el numero de épocas o presentación de datos:
Epochs

Vigilancia

El factor que afecta la selectividad o granularidad de la predicción esta basado en el parámetro de vigilancia. Este será el factor más importante en el controlador. Para lo cual hay que tener en cuenta las siguientes consideraciones:

Base Vigilance (ρ_a) ($0 < \rho_a < 1$)

Se aconseja sea de valor bajo, entendido como un valor muy cercano a cero.

Vigilance (ρ_b) ($0 < \rho_b < 1$)

Se aconseja tenga un valor alto, cercano a 1.

Rho map (ρ_{map}) ($0 < \rho_{map} < 1$)

Si se desea una rápida respuesta se aconseja valores grandes, cercanos a 1. Sin embargo dado que este valor predominantemente actúa sobre la granularidad de la selección pueden existir problemas en ambientes sujetos a ruidos. Aquí hay un compromiso entre la selectividad y rechazo a ruido. Dicho de otro modo por ejemplo si nuestro sistema es muy selectivo y posee un parámetro de vigilancia alto cualquier incremento pequeño en el patrón de entrada puede resultar en una predicción distinta. Sin en cambio, si el parámetro de vigilancia es menor, entonces el sistema será tolerante a ruido y la predicción no se afectará. Este factor realmente dependerá de la naturaleza de los datos a utilizar.

La opción de código de complemento se recomienda utilizar para evitar lo que se conoce como “proliferación de categorías”. El valor de F1 se debe ajustar acorde a los datos que tengamos de entrada y salida.

Para el número de épocas, se recomiendan 2, la experiencia del autor ha demostrado que son suficientes a menos que los datos de entrada sean numerosos en el orden de miles. De otra forma no tiene sentido utilizar mayor numero de épocas ya que el algoritmo converge mas lentamente (mayor a 6 ms). Esto se puede determinar fácilmente con una ejecución del programa observando el archivo de resultados para verificar si la convergencia del algoritmo se da en 1, 2 o n épocas. La convergencia se verifica cuando en cada época subsecuente las predicciones son las mismas.

La capacidad de aprendizaje del controlador esta solamente limitada a la capacidad de memoria en la computadora y la rapidez de convergencia al número de épocas, que normalmente es 1 o 2, lo que hace atractivo utilizar esta herramienta donde se requiere

aprendizaje en línea y en tiempo real. Asimismo, se pueden generar esquemas de descubrimiento y refinamiento del conocimiento cuando se asocia a bases de conocimiento.

Finalmente, también se genera un archivo Simple.dat que es similar al Results.dat, solo que en éste solo aparecen el número consecutivo de patrón y la correspondiente predicción. Este archivo es útil ya que simplifica la integración de los resultados al programa maestro de control. Se tienen planes de trabajar con ActiveX controls para hacer que el controlador pueda ser abierto en un control maestro de manera similar como en MS Word se pueden abrir hojas de calculo en Excel, operar en ellas y volver al documento original. La diferencia en este caso seria que se volveria al programa maestro de control que llama la aplicación.

ANEXO

El controlador neuro-difuso que se presenta esta basado en la teoría ART (Adaptive Resonance Theory). Dicha Teoría fue desarrollada por Stephen Grossberg y Gail Carpenter en la Universidad de Boston. A la fecha se han desarrollado diferentes variaciones en el modelo basadas en el algoritmo original ART-1. La mecánica de un módulo básico ART es como sigue: Consiste de dos subsistemas como se muestra en la Figura A1. El subsistema de atención consta de dos capas de nodos hacia arriba F_1 y F_2 . En una red ART, la información de salida del elemento de procesamiento reverbera hacia atrás y hacia adelante entre las capas. Si una resonancia estable toma lugar, puede ocurrir aprendizaje o adaptación. Por otro lado, el subsistema de orientación está encargado de reinicializar el subsistema de atención cuando ocurre un evento no familiar.

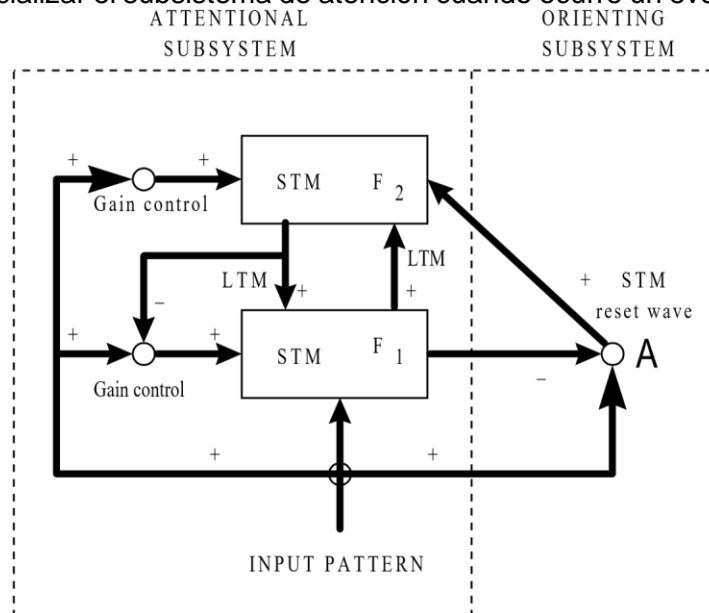


Figura A1 Arquitectura básica ART-1

Un estado resonante puede ocurrir por dos medios. Si la red ha aprendido previamente para reconocer un vector de entrada, entonces un estado resonante toma lugar rápidamente cuando el vector de entrada se presenta. Durante la resonancia, el proceso de adaptación reforzará la memoria del patrón almacenado. Si el patrón de entrada no es reconocido inmediatamente, la red, rápidamente buscará a través de sus patrones almacenados alguno parecido. Si no se encuentra un patrón parecido, la red entrará en resonancia, después de lo cual un nuevo patrón es almacenado por primera vez. Así, la red responde rápidamente a datos aprendidos previamente, permaneciendo aún con la habilidad de aprender, cuando se presentan datos nuevos, por lo tanto se resuelve el dilema denominado estabilidad-plasticidad. La actividad de un nodo en la capa F_1 o F_2 es llamada Memoria de Término Corto (STM) mientras que los pesos adaptivos son llamados memoria de Término Largo (LTM). Los controles de ganancias manejan la presentación discreta de las señales de entrada. Un parámetro de **vigilancia** mide cuanta diferencia es tolerada entre los datos de entrada y los patrones almacenados, los cuales pueden ser usados para controlar la cantidad de categorías del clasificador.

El aprendizaje supervisado es posible también a través de ARTMAP, el cual usa dos módulos ART (ART_a y ART_b) o sus variantes, tales como Fuzzy ARTMAP (FAM) que incorpora un conjunto de operaciones de la teoría Fuzzy, con el objetivo de manejar datos analógicos entre 0 y 1. El controlador fue diseñado tomando como base esta red FAM debido a su capacidad de aprendizaje rápido, típicamente una época (presentación de los datos de entrada una sola vez). La arquitectura de esta red se presenta en la Figura A2.

Aquí aparece también el MapField que es precisamente el campo de donde se toman los resultados de la predicción.

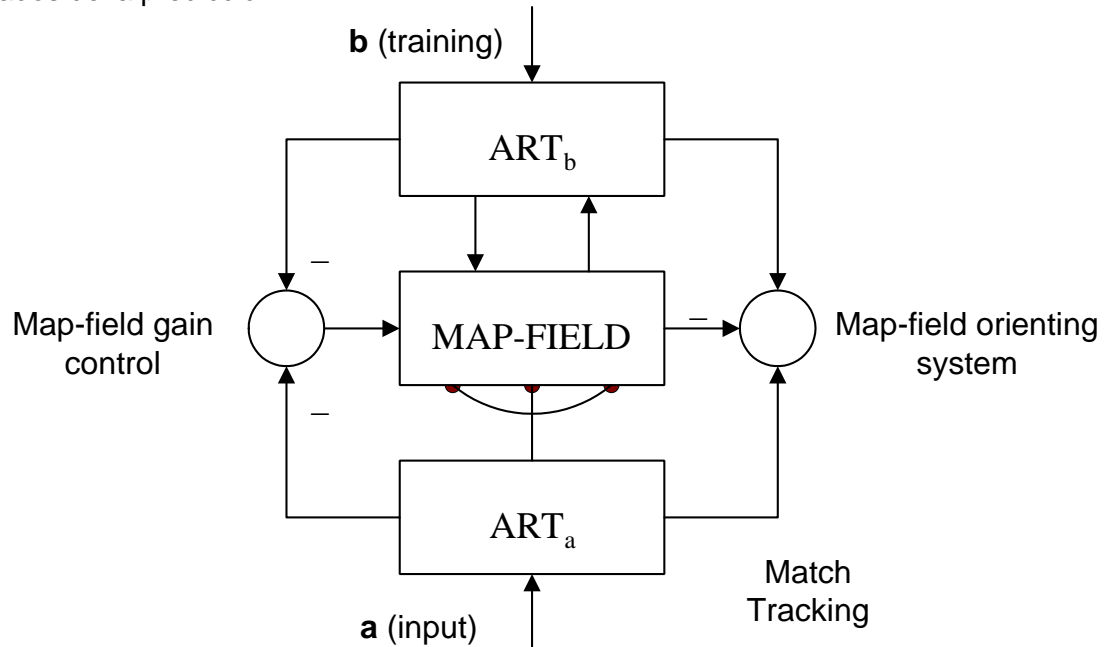


Figura A2 Arquitectura ARTMAP

REFERENCIAS

Gail A. Carpenter; Stephen Grossberg.

Adaptive Resonance Theory (ART).

Michael A. Arbib (ed.): In The Handbook of Brain Theory and Neural Networks.

The MIT Press. (1995) 79-82

Gail A. Carpenter; Stephen Grossberg; Natlya Markunzon; John H. Reynolds; David B. Rosen.

Fuzzy ARTMAP: A neural Network Architecture for Incremental Learning of Analog Multidimensional Maps.

IEEE Transactions on Neural Networks. Vol. 3 No. 5 (1992) 698-713