



Inteligencia Artificial. Revista Iberoamericana  
de Inteligencia Artificial

ISSN: 1137-3601

[revista@aepia.org](mailto:revista@aepia.org)

Asociación Española para la Inteligencia  
Artificial  
España

Domínguez, S.; Campoy, P.  
Arquitectura Neuronal con Aprendizaje Incremental y Creación de Mapas: el Modelo ARM  
Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, vol. 4, núm. 10, verano, 2000,  
pp. 67-75  
Asociación Española para la Inteligencia Artificial  
Valencia, España

Disponible en: <http://www.redalyc.org/articulo.oa?id=92541007>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica  
Red de Revistas Científicas de América Latina, el Caribe, España y Portugal  
Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

# Arquitectura Neuronal con Aprendizaje Incremental y Creación de Mapas: el Modelo ARM

S. Domínguez, P. Campoy

UPM-DISAM, José Gutiérrez Abascal, 2, 28006 Madrid  
e-mail: {sergio, campoy}@disam.upm.es

**Abstract.** En este documento se presenta una nueva arquitectura neuronal capaz de aprender de forma incremental a la vez que realiza una proyección del espacio de entrada sobre una capa de elementos de proceso. El aprendizaje *on-line* permite el diseño de herramientas muy potentes, que pueden ser incluidas dentro de procesos no estacionarios. La proyección del espacio de entrada sobre un mapa de dimensión reducida permite un fácil reconocimiento de situaciones presentadas por el vector de entrada y la evaluación de la proximidad a situaciones límite. Se han dedicado algunos esfuerzos a la creación de modelos que poseyeran ambas capacidades conjuntamente, pero la mayoría de ellos se basan en procesos iterativos y computacionalmente costosos. El nuevo modelo presentado en este documento, llamado ARM (*Adaptive Resonance Map*) consigue este objetivo mediante una arquitectura abierta, permitiendo la inclusión de algoritmos genéricos siempre que cumplan con unos requisitos básicos. Dos algoritmos que cumplen estos requisitos se presentan también, y son probados en problemas sintéticos (aproximación de funciones) y en una aplicación industrial de reconocimiento de patrones visuales, consiguiendo resultados prometedores.

## 1. Introducción

### A. Descripción del problema

La mayoría de los modelos de red neuronal al uso se basan en un ciclo de vida ampliamente conocido: en primer lugar una etapa de entrenamiento, para incluir en la LTM de la red el conocimiento necesario, mediante el uso de algoritmos de adaptación iterativos. Después del entrenamiento, se realiza un paso de validación, para evaluar la habilidad de la red en la generalización, es decir, la extensión de lo aprendido a casos novedosos que puedan presentarse a su entrada. Una vez que la validación arroja resultados satisfactorios, la red está preparada para ser incluida en la aplicación y para realizar la tarea para la que fue diseñada. Cuando una red neuronal se encuadra en semejante ciclo de vida, hay dos suposiciones que se han aceptado desde el principio del problema: primero, el problema que se trata de resolver es estacionario, y segundo, las muestras utilizadas en el proceso de entrenamiento incluyen una representación completa de todos los posibles casos que el problema puede presentar. La primera suposición, de estacionariedad, significa que todos los parámetros

implicados en el problema van a permanecer sin cambios; con el tiempo, las propias características del problema no van a variar durante toda la vida de la aplicación. Incluso una red entrenada para predecir el valor de una determinada variable a medida que el tiempo pasa fallará si alguno de los parámetros que la definen cambia.

La práctica nos enseña que los problemas en la vida real son normalmente no estacionarios, puesto que suelen verse afectados por el tiempo, máxime cuando se habla de procesos industriales donde la maquinaria y las condiciones del entorno juegan un papel crucial.

La segunda suposición mencionada, disponibilidad de un conjunto de muestras de entrenamiento representativo, implica un proceso de entrenamiento previo a la vida útil de la red, en el cual una gran cantidad de muestras representativas de todo el problema debe ser introducida al sistema de aprendizaje. En muchos problemas de clasificación en un entorno industrial, las muestras procedentes de alguna o de varias de las clases relevantes son

escasas, poco representativas, o simplemente no se encuentran disponibles, impidiendo por tanto la consistencia del posterior paso de clasificación.

En estos casos, dado un vector de entrada, el significado de la salida de la red puede cambiar a medida que el problema evoluciona, debido a que los valores de entrada pueden verse afectados en el tiempo. En este caso, en entrenamiento inicial de la red pierde su validez, dado que el problema que la red estaba resolviendo inicialmente se ha modificado. De esta forma, el ciclo de vida de la red debe ser modificado para mantener su capacidad de aprendizaje mientras continúa resolviendo su tarea, es decir, ser capaz de aprender mientras el propio problema evoluciona. En otras palabras, la red debe mantener su estado plástico, su capacidad de modificar los contenidos de su LTM durante toda su vida [2].

La inclusión de algún mecanismo que permita la adaptación de la red a la vez que se encuentra realizando su tarea introduce algunas implicaciones en absoluto triviales en el aprendizaje y utilización de la red, problemas que deben ser completamente resueltos para conseguir un diseño preparado para afrontar problemas no estacionarios. La respuesta más conocida para esta problema se ha formulado mediante la teoría de resonancia adaptativa, redes ART, que solucionó el llamado problema de estabilidad-plasticidad, que es, en pocas palabras, una generalización de la cuestión expuesta anteriormente [2]. Hablando más propiamente, este dilema introduce otro aspecto en el problema. Si la red necesita aprender a la vez que se encuentra trabajando, debe existir alguna protección para el conocimiento adquirido con anterioridad, puesto que se encuentra adquiriendo nuevo conocimiento para resolver el problema a medida que evoluciona, y éste debe ser almacenado en un espacio finito. Este es el caso típico de un problema en crecimiento, más que sometido a cambios. Si la estructura de la red es finita de por sí, el conocimiento que es capaz de almacenar también lo es. En el caso límite, alguna zona de memoria podría ser borrada para almacenar nuevo conocimiento, conduciendo a una degradación de la LTM, dado que la red perdería su capacidad para resolver el problema inicial.

Existen algunos modelos con un ciclo de vida del estilo aprendizaje-validación-trabajo, y algunas de sus capacidades han probado un rendimiento óptimo. Uno de estos modelos con una amplia aplicación y un rendimiento excelente es el mapa autoorganizativo de Kohonen, los SOM's [10], capaces de proyectar el espacio de entrada sobre una matriz de elementos de proceso, llamada el mapa de salida. Esta proyección se realiza de acuerdo con

dos propiedades débiles: vecindad y ordenación. La primera hace referencia al hecho de que dos puntos que aparezcan cerca en el mapa de salida deben encontrarse próximos en el espacio de entrada, y la segunda propiedad establece que dos puntos que se encuentran cerca dentro del espacio de entrada deben aparecer cerca en el mapa de salida. Estas propiedades han sido calificadas como débiles puesto que su cumplimiento no es obligatorio para ningún par de puntos ni en el mapa ni en el espacio de entrada. Esta falta de precisión proviene del hecho de que el aprendizaje no viene guiado por ningún criterio de ordenación, sino por uno de proximidad. El aprendizaje tiende a hacer a los vecinos más próximos cada iteración, pero esto no es garantía de conseguir un mapa ordenado. Sin embargo, la ordenación media en el mapa completo es suficiente para conseguir la formulación de hipótesis útiles en cuanto a la posición relativa de regiones significativas en el espacio de entrada.

El objetivo del trabajo presentado en este documento se ha orientado a conseguir aunar en un modelo la adaptación en línea con la proyección del espacio de entrada [4].

Como se establece en párrafos anteriores, la adaptación no iterativa y la proyección son problemas adecuadamente resueltos por separado. La dificultad viene cuando se trata de mezclar ambas características en un único modelo. Esto se debe al hecho de que el proceso de aprendizaje es completamente distinto para ambos tipos de modelos. Para el primero, cuya arquitectura más conocida es la ya mencionada ART, donde el aprendizaje es un proceso no iterativo, y la adaptación se mezcla con la recuperación en el ciclo de vida de la red, el aprendizaje se consigue mediante la creación de nuevos prototipos, que se asocian a una estructura constructiva básica que se replica para albergar el nuevo conocimiento. De esta forma, la degradación del conocimiento adquirido previamente se evita dado que el nuevo prototipo se almacena en una estructura distinta y nueva. Todos estos pasos se realizan mediante una arquitectura de red compleja y recurrente.

En cuanto a la segunda característica mencionada, es decir, la creación de mapas mediante la proyección del espacio de entrada, las técnicas usadas para el entrenamiento son totalmente distintas a las empleadas en el modelo ART. Este hecho puede comprobarse estudiando el entrenamiento de los SOM's, la red más conocida que incluye esta característica. En primer lugar, la propia topología de la red es distinta, y se fija antes del entrenamiento, en una fase de diseño previa, por lo que no hay adición de nuevas estructuras durante

la recuperación. El esquema de conexión entre los elementos de proceso y las distintas capas se decide de forma previa, asimismo. Por lo tanto, no es posible añadir cambios en la topología de la red una vez que el entrenamiento ha terminado, y no existe la posibilidad de añadir nuevo conocimiento a la memoria ya existente al acabar esta fase. Además, la formación del mapa se consigue mediante un proceso iterativo, fuera de línea, y computacionalmente costoso.

### B. Estado del arte

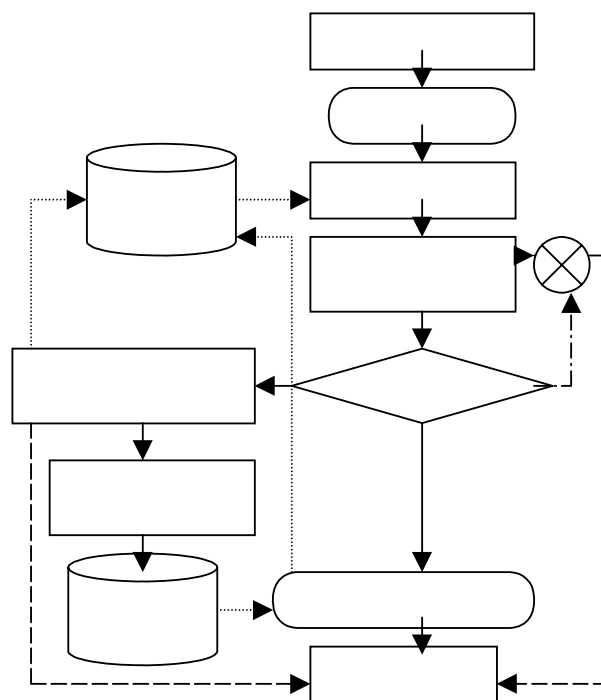
El objetivo de la nueva arquitectura es la unión en un solo modelo de red neuronal de las características de formación de mapa y de adaptación en línea. Algunos trabajos en este sentido se han llevado a cabo con anterioridad al que aquí se describe: Fritzke desarrolló un modelo en el que un mapa se crea dinámicamente mediante la adición de nuevos nodos a la estructura actual [8][9]. Pero en este modelo, todo el conocimiento pertinente al problema debe estar disponible al comienzo del proceso de entrenamiento, y la adición de nuevos nodos al mapa se decide después de un proceso iterativo y computacionalmente costoso, durante el que se calcula una medida acumulativa de error para cada nodo de la red.

Blackmore y Miikulainen desarrollaron un modelo basado en los mismos principios [1], en el que la característica adicional que se presenta sobre el modelo de Fritzke es la posibilidad de realizar una representación gráfica del estado de la red en un plano, lo que la hace muy apropiada para tareas de visualización. Existen otras diferencias de menor entidad, como la necesidad de utilizar heurísticos en este segundo modelo, y la estructura básica utilizada para la conformación del mapa: el modelo de Fritzke utiliza como estructura básica el triángulo, mientras que este segundo modelo utiliza cuadrados, que pueden degenerar a rectas a medida que el aprendizaje progresa. En ambos casos, el uso de procesos iterativos en la fase de aprendizaje hace el entrenamiento muy largo y costoso, y fuerza a la disponibilidad de un conjunto de aprendizaje completo, en el sentido de dar una representación global del problema, de forma que el esquema de entrenamiento-recuperación pueda ser usado.

La conclusión es que ambas filosofías, adaptación en línea y creación de mapas, no han sido compaginadas con éxito para aplicaciones industriales con fuertes requisitos de temporalidad, al menos hasta la fecha.

## 2. Descripción del nuevo modelo: la arquitectura ARM

Con objeto de sintetizar un modelo neuronal que aglutine las características de adaptación en línea y creación de mapas, la arquitectura ARM (*Adaptive Resonance Map*) se ha desarrollado en forma de especificaciones generales, más que de estructuras neuronales puras, de forma que el modelo quede abierto a implementaciones alternativas [6]. El esquema formulado, que se presenta en la figura 1, se explica en los siguientes párrafos.



En la figura 1 las líneas continuas representan la secuencia de pasos desde que la entrada se presenta a la red hasta que ésta ofrece un resultado a la salida. La línea discontinua fina representa los pasos para completar la memoria almacenada con nuevo conocimiento aprendido del entorno, así como el flujo de información desde la memoria hacia los puntos en los que es necesario disponer de él. La línea discontinua a trazos largos representa el flujo de información necesario para la construcción de la salida. En cuanto a la línea discontinua mixta (corto largo) representa el flujo de la información de control, necesario para bloquear el paso de la información facilitada por la red hasta la salida cuando dicha información ha sido rechazada en el paso de validación. Los bloques con vértices agudos representan pasos necesarios para alcanzar la solución, mientras que los bloques con esquinas redondeadas representan pasos opcionales, no necesarios pero aconsejables. Las cajas rómbicas

representan pasos en los que se toma alguna decisión. Finalmente, los bloques cilíndricos representan el almacenamiento de memoria, en los que la información relativa a los prototipos y a su posición y conexión en el mapa son almacenados. La red ARM funciona como sigue. Cuando una muestra de entrada llega a la red, debe ser codificada de la misma forma en la que los prototipos se encuentran almacenados en la red, de forma que resulte comparable con ellos. Este paso es por tanto opcional, puesto que los prototipos pueden encontrarse almacenados en la memoria de la red tal como son, sin transformación alguna entre las coordenadas del espacio de entrada y la representación interna del conocimiento en la red. Una vez que los datos de entrada están codificados de forma que sean compatibles con la representación de la memoria de la red, el siguiente paso es la comparación entre esta entrada y los prototipos ya almacenados. A partir de esta comparación se obtendrá un ganador, que es el candidato a servir de representación a la entrada, o dicho de otra forma, el prototipo del cluster conocido más similar a la entrada. Como en las redes ART, este candidato debe ser validado antes de ser aceptado para servir como clasificación de la muestra de entrada. Este paso de validación puede terminar de dos formas distintas. Si el candidato es aceptado, será incluido como parte de la respuesta final. En caso contrario, el ganador no formará parte de dicha respuesta, y la red se verá obligada a crear un nuevo prototipo, de forma que las entradas procedentes de la misma región que la actual se vean propiamente representadas. La validación se realiza mediante la umbralización del valor de la comparación de la entrada con el candidato [2]. Si se ha creado un nuevo prototipo, debe ser incluido en la estructura actual del mapa; los algoritmos utilizados para realizar esta inserción sin procesos iterativos serán presentados más adelante.

Se ha incluido también un paso opcional en el que las posiciones de los prototipos en el mapa pueden ser actualizadas. Esta actualización se lleva a cabo para aumentar la precisión del mapa y para optimizarlo, es decir, para que ofrezca una mejor cobertura de las regiones de interés con los prototipos ya existentes. Los cambios en las posiciones de los prototipos deben ser muy pequeñas, puesto que grandes desplazamientos en sus posiciones pueden conducir a una variación en su significado, lo que conduciría a clasificaciones erróneas de las muestras asociadas en principio a dicho prototipo.

Nótese que para la selección y validación del prototipo candidato a representar a la muestra de entrada se ha supuesto que el conocimiento

almacenado en la red está expresado como un conjunto de prototipos, o paradigmas de clase, y que la entrada es un patrón expresado en los mismos términos, es decir, coordenadas con el mismo significado, o que al menos existe una transformación que haga su expresión compatible con la de la memoria de la red. Esta representación del conocimiento se hereda de modelos que realizan proyección del espacio de entrada en mapas, y en los que los pesos de conexión de la capa del mapa con la entrada codifican la posición de los prototipos en el espacio de entrada [1][8][9][10][11].

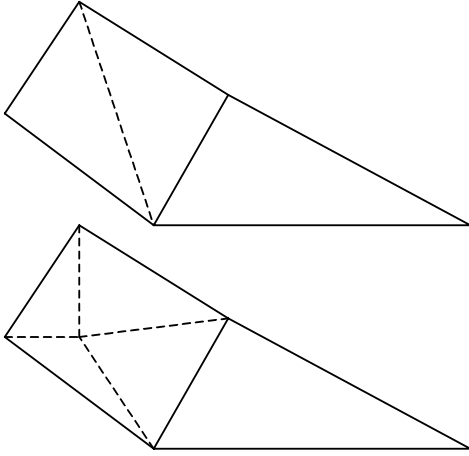
Los modelos sin un paso de validación de la salida generan alguna respuesta incluso en el caso de que ésta carezca de sentido por proceder de una entrada desconocida, y por lo tanto no pueda ser reconocida. Por el contrario, el modelo ARM reconoce esta novedad y crea un nuevo prototipo para clasificar nuevas ocurrencias del mismo tipo de entrada. La detección de novedad acarrea, como consecuencia lógica, la creación de un prototipo nuevo. Para incluir este nuevo prototipo en el mapa, los modelos existentes deben seguir un entrenamiento completo desde cero, siendo necesaria una colección representativa de muestras, en el sentido de que debe incluir muestras procedentes de todas las regiones significativas del espacio de entrada. El modelo ARM es capaz de incluir este nuevo conocimiento en un solo paso, sin necesidad de realizar procesos iterativos ni de repetir el entrenamiento.

### **3. Implementación del modelo**

El criterio utilizado para la inclusión de nuevos prototipos en la estructura del mapa es el mismo utilizado en el entrenamiento de los SOM: la suposición de que potenciando la vecindad localmente el algoritmo de entrenamiento puede generar ordenación global en el mapa. Esta ordenación se consigue a base de pequeños desplazamientos de los prototipos vecinos para hacerlos más próximos. Después de una cantidad variable de iteraciones, que depende de la complejidad del problema, el resultado final es un mapa donde los elementos vecinos representan puntos cercanos en el espacio de entrada.

En este documento se presentan dos algoritmos que hacen uso de esta idea para insertar nuevos prototipos en el mapa, a la vez que respetan la restricción de aprendizaje en línea. Ambos algoritmos hacen uso de criterios topológicos que relacionan elementos la estructura actual del mapa con el nuevo prototipo, siendo ambos no iterativos. Ambos serán explicados en las secciones siguientes.

Asimismo, ambos asumen que el elemento básico para la construcción del mapa es el triángulo, de forma que los algoritmos presentados se basarán en la división de triángulos ya existentes para encontrar la mejor representación del espacio de entrada. La división del triángulos se realiza mediante la sustitución de alguna de sus aristas por dos nuevas, lo que implica la sustitución de un triángulo por otros dos nuevos en cada inserción, o por cuatro, si es que la arista sustituida es compartida por dos triángulos. La inicialización de los algoritmos se lleva a cabo simplemente mediante la creación de un mapa inicial con un solo triángulo a base de los tres primeros prototipos creados por la red. El mecanismo de inserción queda reflejado en la figura 2.



#### A. Reducción de la distancia Media de vecindad

El primer algoritmo de aprendizaje incremental se basa en la suposición de que la reducción de la distancia cuadrática entre elementos conectados (vecinos) acusa la ordenación óptima en el sentido de las distancias de vecindad largas son penalizadas, mientras que las cortas se encuentran favorecidas. Si el algoritmo tiende a reemplazar distancias de conexión largas debido a esta penalización, solo las distancias cortas sobrevivirán, por lo que un par de vecinos tenderá a estar tan próximo como sea posible. Este algoritmo, por lo tanto, se basa en un criterio de proximidad, igual que el entrenamiento de los SOM, pero consiguiendo la cercanía entre elementos vecinos en un solo paso no iterativo.

La formulación de este algoritmo es bastante simple: sea la distancia cuadrática media en el mapa,  $C(t)$ , para el instante  $t$ :

$$C(t) = \frac{\sum_{\forall (P_i, P_j)} |P_i - P_j|}{s}$$

donde  $(P_i, P_j)$  representa una pareja de puntos conectados,  $s$  es el número de pares conectados existentes en el mapa y:

$$|P_i - P_j| = \sum_{k=1}^n (p_{ik} - p_{jk})^2$$

es la distancia cuadrática entre un par de miembros del mapa, siendo  $n$  la dimensión de cada vector y  $p_k$  sus componentes.

Dadas las anteriores definiciones, la inclusión de un nuevo elementos originará la creación de  $a$  (siendo  $a=3$  para un borde perteneciente a un triángulo y  $a=4$  para bordes compartidos por dos triángulos) nuevas conexiones, y la eliminación de una nueva, como se explicó en los párrafos anteriores. La creación de estas nuevas conexiones y la eliminación de la antigua supone un cambio en el valor de la distancia media de vecindad; para el instante  $t+1$ , tras la inserción, pasa a valer  $C(t+1)$ :

$$C(t+1) = C(t) \frac{s}{s+a-1} + \frac{\sum_{\forall l} |P_l - P_{N+1}| - |P_u - P_v|}{s+a-1}$$

donde el primer término al lado de la derecha es la corrección de la media anterior según el nuevo número de conexiones,  $a$  es el nuevo número de conexiones que aparecen tras la inserción del nuevo nodo,  $l$  es el conjunto de nodos conectados al nuevo elemento,  $N+1$  es el número de elementos en el mapa en el instante  $t+1$  y  $(P_u, P_v)$  es la conexión que ha sido reemplazada tras la inserción.

Dado que el primer término de la derecha es fijo, puesto que  $C(t)$  es un dato y  $s$  y  $a$  son constantes, no toman parte en la reducción (o aumento mínimo) de la distancia media de conexión, de forma que para la optimización el término que ha de ser minimizado es:

$$\tilde{C}(t+1) = \sum_{\forall l} |P_l - P_{N+1}| - |P_u - P_v|$$

La posición de inserción en el mapa será, por tanto, la que haga mínimo este término.

#### B. Selección de la arista más cercana

Este criterio se basa en la selección de la arista que se encuentre más próxima a la posición del nuevo prototipo. La expresión para la evaluación de este criterio es:

$$D(P_{N+1}, (\overline{P_i, P_j})) = \begin{cases} d(P_{N+1}, (\overline{P_i, P_j})) \\ d(P_{N+1}, P_i) \text{ if } P_i \text{ is the closest} \\ d(P_{N+1}, P_j) \text{ if } P_j \text{ is the closest} \end{cases}$$

donde  $D(.)$  representa la distancia entre el nuevo prototipo y una arista definida por dos de los prototipos ya existentes en el mapa. En el primer caso de la definición, la perpendicular al segmento

$(\overline{P_i, P_j})$  que contiene a  $P_{n+1}$  corta al primero entre ambos puntos. En caso contrario,  $D(.)$  se define como la distancia entre  $P_{N+1}$  y el prototipo más cercano de los dos que definen el segmento (casos segundo y tercero). De acuerdo con este criterio, el segmento que minimiza esta definición de distancia es el que resulta seleccionado para ser reemplazado.

Este algoritmo permite la ordenación partiendo de una aproximación distinta que en el caso de la reducción de la distancia de conexión media. En este caso, la cercanía del nuevo prototipo no se evalúa midiendo la distancia a los otros prototipos, sino a aristas ya existentes. Por lo tanto, la ordenación proviene del hecho de que las conexiones seleccionadas para el nuevo prototipo son aquellas que introducen la mínima distorsión posible en la configuración actual del mapa, o en otras palabras, este algoritmo tiende a hacer que la nueva inserción deje al mapa tan inalterado como sea posible, puesto que el mejor caso se da cuando el prototipo cae dentro de una arista ya existente, por lo que el cambio, en cuanto a esa arista, es su simple división en dos tramos, por lo que el nodo se añade sobre la estructura presente, sin modificarla.

#### 4. Resultados de la nueva arquitectura

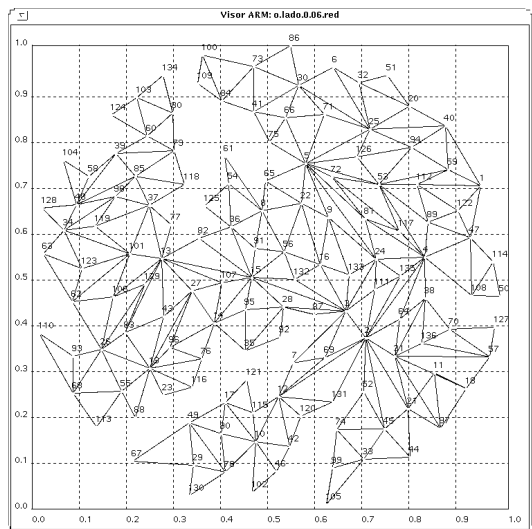
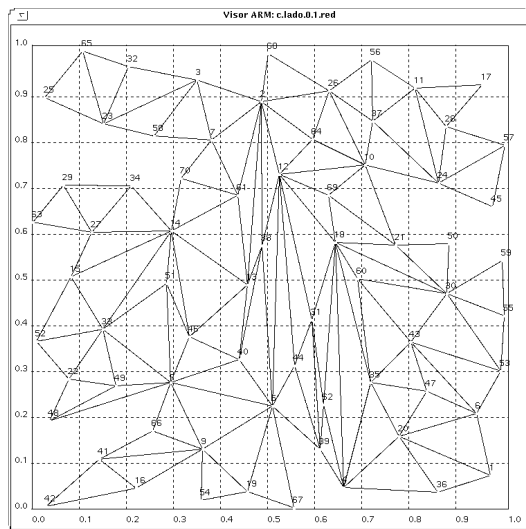
El modelo ha sido probado bajo dos puntos de vista diferentes. En la primera fase, se le ha sometido a la prueba de resolver problemas de aproximación de funciones, donde debía componer un mapa que reflejase funciones de distribución de probabilidad predefinidas sobre un espacio de entrada bidimensional. La evaluación de la solución dada a estos problemas, por el análisis del mapa plano resultante, se probó la habilidad del mapa para la resolución de tales tareas. En una segunda etapa,

una red creada siguiendo el modelo ARM ha sido incluida en una aplicación industrial de inspección, donde su papel ha sido la clasificación de patrones provenientes de un sistema de visión por computador inspeccionando banda de acero a medida que se procesa.

#### A. Resultados en problemas sintéticos

El modelo fue sometido a la prueba de resolver un problema de aproximación de funciones, donde las muestras de entrenamiento provenían de una selección aleatoria de algunas distribuciones uniformes bidimensionales. Algunos resultados obtenidos en esta tarea se muestran en la figura 3.

En la figura 3 se puede observar que existen algunas concavidades en el perfil del mapa. Estas concavidades, aparentemente no se observan en los mapas de Kohonen ni en los otros algoritmos con crecimiento mencionados en este documento [1][8][9][10][11], cuando resuelven problemas de aproximación de funciones en 2D. Su presencia en este caso es consecuencia directa del crecimiento del mapa por la adición de nuevas aristas cuando el nodo a añadir se encuentra fuera de los límites actuales del mapa. El efecto de estas concavidades es susceptible de un análisis más detallado. En primer lugar, el error de cuantización no se ve afectado por su aparición, dado que dichas concavidades aparecen como consecuencia del algoritmo de conexión y no del de creación de nodos, que es el que fijaría dicho error; de esta manera, la calidad de la proyección en términos de error no se ve afectada. Por otro lado, cuando cualquiera de las redes mencionadas trabaja en espacios de entrada de dimensión elevada y realizando proyecciones bidimensionales, las concavidades siempre aparecen: piénsese en una



hoja de papel aproximando el volumen de un jarrón. La hoja debería doblarse muchas veces para conseguir una aproximación a un volumen, dado que es un objeto plano. Obviamente, este proceso de doblado resulta en la aparición de muchas concavidades, sea cual sea el modelo neuronal utilizado. En otras palabras, dado que los problemas reales (de alta dimensión) ocasionan la aparición de concavidades en la proyección, no tiene sentido práctico realizar modificaciones en el algoritmo para evitar su aparición en problemas de dimensión reducida, simplemente para lograr una mejor apariencia en tareas de demostración.

A partir de estos resultados se puede deducir que el modelo presenta una alta capacidad para la aproximación de funciones. La calidad de estos resultados revela que la solución dada es tan buena como la de los algoritmos de aprendizaje fuera de línea e iterativos. Las principales ventajas del modelo ARM propuesto son, por lo tanto, su capacidad para actualizar los contenidos de su LTM a medida que muestras desconocidas llegan a la entrada, evitando cualquier ciclo iterativo y costoso para conseguir esta actualización. Por otra parte, pero simultáneamente, la red es capaz de hacer este nuevo conocimiento compatible con el antiguo, de tal manera que un mapa de proyección del espacio de entrada puede ser almacenado en la red, e introduciendo nuevos prototipos que son incorporados a la red completamente en línea, en el lugar más apropiado del mapa, de tal forma que se mantiene su semejanza con el espacio de entrada hasta el punto en el que es actualmente conocido.

## B. Resultados en una tarea de inspección industrial

Como se ha mencionado con anterioridad, el modelo propuesto ha sido incluido en una aplicación industrial, donde su cometido ha sido realizar la clasificación de patrones correspondientes a defectos superficiales encontrados sobre una banda de acero mediante un sistema de visión por computador [13]. Esta tarea de inspección se realiza a medida que el acero se está procesando: 18 características visuales son extraídas en línea por el sistema de visión por computador y enviadas al clasificador neuronal; estas características describen fundamentalmente la geometría del objeto observado y la distribución espacial de los píxeles que lo componen [6]. Los requisitos principales de esta inspección son detección en línea, no estacionariedad del problema y evaluación del grado de certeza de la clasificación. La estructura ARM que se propone se ha probado en el reconocimiento de seis tipos distintos de defectos superficiales del acero, cada uno con tres niveles distintos de severidad. El clasificador fue evaluado con

aproximadamente 500 muestras reales de defectos, recogidas de la línea de producción mediante el sistema de visión por computador. Los resultados obtenidos se recogen en la tabla 1, donde la primera columna numérica muestra el número de muestras clasificadas dentro de cada categoría y la segunda la cantidad relativa sobre el total. Una característica importante que fue probada con éxito es la posibilidad de utilizar la información topológica suministrada por el modelo mediante la creación del mapa para evaluar el riesgo de ofrecer una clasificación equivocada para una muestra de entrada dada. Esta evaluación se basa en la proximidad del nodo ganador al borde de clase más cercano en el mapa, y ha demostrado su utilidad, puesto que las clasificaciones equivocadas son siempre etiquetadas como dudosas.

Reconocidas	Seguras	217	55.0%	Bien	217	100%
				Mal	0	0%
	Dudosas	70	17.5%	Bien	68	97.0%
				Mal	3	3.0%
Desconocidas		109	27.5%			

**Tabla1. Resultados del modelo ARM**

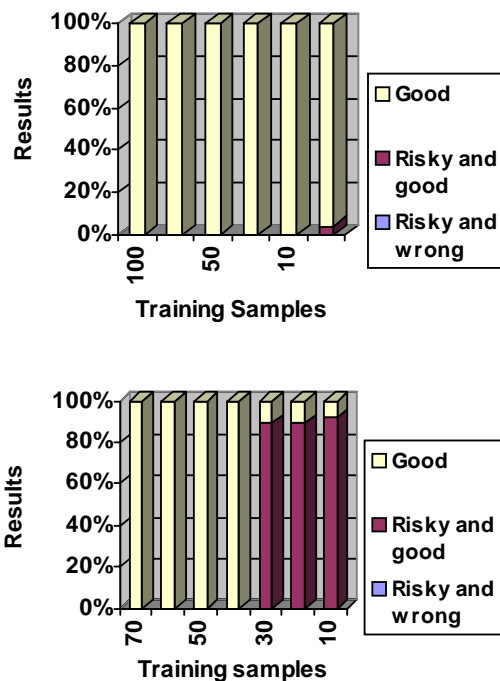
La capacidad del modelo para ofrecer clasificaciones correctas tras un corto entrenamiento puede ser evaluada con la siguiente prueba. Un clasificador se ha probado con muestras representativas de seis defectos, pero uno de ellos con un conjunto reducido; esta reducción se ha aplicado progresivamente, para evaluar la degradación del comportamiento de la red cuando el conocimiento relativo a una de las clases se va reduciendo. Esta prueba se ha repetido para distintos tipos de defecto, reduciendo en cada caso el conocimiento relativo al mismo en la red.

Los resultados de esta prueba se muestran en la figura 4, en la que se muestra el buen rendimiento alcanzado con el clasificador ARM, capaz de ofrecer clasificaciones libres de error incluso cuando el conjunto de entrenamiento (conocimiento relativo a una clase adquirido previamente) es limitado a diez muestras, que son mostradas a la red solo una vez. El único efecto que se puede observar es, para el segundo caso mostrado, el etiquetado de resultado dudoso para casi todas las muestras del conjunto de test cuando el número de muestras conocidas previamente baja por debajo de 30. Pero incluso en este caso, todas las conclusiones alcanzadas son correctas.

Los resultados demuestran la capacidad de la red para realizar clasificaciones complejas. La red es capaz de ofrecer una respuesta de alta calidad solo después de haberle sido mostrada una pequeña cantidad de muestras, de forma que el lapso de



tiempo entre ausencia total de conocimiento del problema y resultados útiles es muy corto; en las primeras etapas de funcionamiento en línea el sistema aprende, creando prototipos para describir las clases, y solo tras una reducida cantidad de muestras el sistema ya ha adquirido suficiente conocimiento para ofrecer buenas clasificaciones. Por lo tanto, este modelo puede ser considerado como una herramienta muy potente en aplicaciones donde la adaptación en un tiempo reducido es crítica (procesos no estacionarios) y el tiempo de desarrollo (utilizado para diseñar y ajustar el clasificador) debe ser corto también, consiguiendo buenos resultados en la clasificación tras un reducido número de muestras procedentes del entorno.



## 5. Conclusiones

En este documento se presenta un nuevo modelo neuronal, que posee la capacidad de crear mapas de proyección en línea que representan la distribución de clases en el espacio de entrada sobre una capa de elementos de proceso dentro de la estructura de la red funcionando en tareas no estacionarias. Esta proyección en línea se consigue mediante algoritmos de aprendizaje incremental, de tal forma que las clases son creadas a medida que las muestras pertenecientes a dichas clases son presentadas a la red. Dos algoritmos para conseguir esta adaptación en línea son presentados. Como consecuencia de todo lo anterior, el tiempo de creación del mapa, en

comparación con otros modelos iterativos, es reducido drásticamente.

Se presentan resultados que prueban la calidad de las soluciones que el modelo ofrece en problemas sintéticos y en aplicaciones industriales. Los resultados del modelo en la aplicación industrial prueban su capacidad para resolver problemas de clasificación complejos, donde las condiciones son de no estacionariedad. Además, la topología del mapa se utiliza para deducir información útil relativa a la fiabilidad de la clasificación, lo que conduce a la confirmación por el usuario de las clasificaciones calificadas como dudosas. La capacidad para trabajar en entornos no estacionarios se aprovecha para introducir el clasificador en línea incluso cuando no ha adquirido conocimiento alguno previamente, de forma que la adaptación se consigue mediante el aprendizaje de muestras reales dentro de la línea.

## 6. Agradecimientos

Los autores desean hacer patente su agradecimiento a la Comisión Europea para el Carbón y el Acero (CECA) por la financiación aportada para la realización del proyecto de inspección industrial, y a Aceralia (antigua ENSIDESA) por su colaboración en el mismo y por facilitar sus instalaciones en Avilés (Asturias) para la implementación y puesta en marcha del sistema de clasificación aquí descrito.

## 7. Bibliografía

- [1] Blackmore, J., Mikkulainen, R. (1992). Incremental Grid Growing: Encoding High Dimensional Structure into a Two Dimensional Feature Map. Technical Report AI92- 192, University of Texas at Austin.
- [2] Carpenter, G.A., Grossberg, S. (1987). A Massively Parallel Architecture for a Self-Organising Neural Pattern Recognition Machine. Computer Vision, Graphics and Image Processing, Vol. 37, pp. 54-115.
- [3] Carpenter, G.A. (1989). Neural Network Models for Pattern Recognition and Associative Memory. Neural Networks, Vol. 2.
- [4] Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., Rosen, D.B. (1992). Fuzzy ARTMAP: Supervised Real Time Learning and Classification of Nonstationary Data by a Self Organising Neural Network. Neural Networks, Vol. 4, pp. 161- 182.
- [5] Dominguez, S., Campoy, P., Aracil, R. (1994) A Neural Network Based Quality Control System for Steel Strip Manufacturing. Preprints of the Symposium on Artificial Intelligence in Real Time Control, AIRTC'94.
- [6] Dominguez, S. (1996). Arquitectura Neuronal de Aprendizaje Continuo con

Crecimiento Topologicamente Coherente.  
Aplicacion al Reconocimiento de Patrones  
Visuales. Doctoral Thesis. 1996.

- [7] Elman, J.L. (1991). Incremental Learning or the Importance of Starting Small. Center for Research in Language, Technical Report 9101, University of California at San Diego.
- [8] Fritzke, B. (1991). Unsupervised Clustering with Growing Cell Structures. Proceedings of the International Joint Conference on Artificial Neural Networks, Vol. II, pp. 531 -536.
- [9] Fritzke, B. (1993). Growing Cell Structures. A Self Organising Network for Unsupervised and Supervised Learning. International Computer Science Institute, TR-93-026.
- [10] Kohonen, T. (1990). The Self Organising Map. Proceedings of the IEEE. Vol. 78, N° 9, pp.1464-1480.
- [11] Kohonen, T. (1995). Self Organising Maps. Springer-Verlag.
- [12] Nolfi, S., Parisi, D. (1992). Growing Neural Nets. Proceedings of Artificial Life m, Santa Fe, New Mexico.
- [13] Obeso, F., González, J.A., Brown, A., Domínguez, S., Campoy, P., Mochón, J. (1997). Intelligent On Line Surface Inspection on a Skinpass Mill. Iron and Steel Engineer, September 1997.
- [14] Weng, J., Ahuja, N., Huang, T.S. (1992). Cresceptron: A Self Organising Neural Network Which Grows Adaptively. Proceedings of the International Joint Conference on Neural Networks, Vol. I, pp 576-581.
- [15] Campoy, P., Domínguez, S., Platero, C. (1998). Aplicación de las Redes Neuronales a la Inspección Visual Automatizada. Automática e Instrumentación, nº 290, pp. 87-95.