

Diseño de redes neuronales con aprendizaje combinado de retropropagación y búsqueda aleatoria progresiva aplicado a la determinación de austenita retenida en aceros TRIP^(*)

I. Toda-Caraballo*, C. Garcia-Mateo* y C. Capdevila*

Resumen

A partir de los años noventa, el interés que los aceros TRIP despiertan en la industria ha producido un incremento considerable de su estudio y aplicación. En este trabajo, aprovechando la flexibilidad que las redes neuronales proporcionan para la modelización de propiedades complejas, se ha abordado el problema de la determinación de la austenita retenida en los aceros TRIP. Una combinación de dos algoritmos de aprendizaje (retropropagación y búsqueda aleatoria progresiva) de la red neuronal ha permitido crear un modelo que predice la cantidad de austenita retenida en aceros multifase con bajo contenido en aluminio y silicio en función de los parámetros de procesado.

Palabras Clave

Redes neuronales; Búsqueda aleatoria progresiva; Algoritmo de retropropagación; Austenita retenida; TRIP.

Neural network design with combined backpropagation and creeping-random-search learning algorithms applied to the determination of retained austenite in TRIP steels

Abstract

At the beginning of the decade of the nineties, the industrial interest for TRIP steels leads to a significant increase of the investigation and application in this field. In this work, the flexibility of neural networks for the modelling of complex properties is used to tackle the problem of determining the retained austenite content in TRIP-steel. Applying a combination of two learning algorithms (backpropagation and creeping-random-search) for the neural network, a model has been created that enables the prediction of retained austenite in low-Si / low-Al multiphase steels as a function of processing parameters.

Keywords

Neural Networks; Creeping-random-search; Back propagation algorithm; Retained austenite; TRIP.

1. INTRODUCCIÓN

El desarrollo de aceros multifase tiene una gran importancia en la industria del automóvil, ya que permite desarrollar componentes de peso ligero con excelentes propiedades mecánicas. Estos aceros presentan una microestructura compleja constituyente de fases blandas (ferrita, austenita) y duras (martensita, bainita), cuya combinación ofrece un balance óptimo entre la resistencia y la ductilidad. En este contexto tiene especial relevancia la fase austenita, ya que es una fase metaestable a temperatura ambiente que se transformará en martensita cuando se aplica una deformación externa. Este proceso de transformación

por deformación se denomina efecto TRIP (Plasticidad Inducida por Transformación-*Transformation Induced Plasticity*) y mejora de manera apreciable las propiedades mecánicas del acero.

La creación de estos aceros y su procesado incluye un número alto de variables y ofrece unos resultados suficientemente diversos que hace difícil la optimización de sus fases (austenita retenida en este caso) para mejorar sus características. Tradicionalmente, la modelización de propiedades metalúrgicas hace uso de redes neuronales que son capaces de sintetizar el conocimiento experimental a través de una descripción matemática que permiten obtener buenas aproximaciones de propiedades físicas de interés

^(*) Trabajo recibido el día 24 de Mayo de 2009 y aceptado en su forma final el día 1 de Julio de 2010

*Grupo Materialia - Dep. Metalurgia Física. Centro Nacional de Investigaciones Metalúrgicas (CENIM). Consejo Superior de Investigaciones Científicas (CSIC), Av. Gregorio del Amo, 8, 28040 Madrid.

para el desarrollo de nuevos materiales. Dicha descripción matemática pasa frecuentemente por el uso de perceptrones multicapa que aseguran una manera sencilla y elegante de comprimir la información^[1-3]. Sin embargo, la gran dispersión y variabilidad de los datos recopilados en la bibliografía^[4-30] obliga a definir la red neuronal de un tamaño suficientemente grande para comprimir este conocimiento. En el proceso de aprendizaje u optimización del error en los parámetros de salida, una red neuronal de estas características con gran dispersión en la base de datos puede ralentizar y dificultar la búsqueda de soluciones de la red neuronal. El clásico algoritmo de la retropropagación es sensible en la obtención de mínimos locales, ya que es no es más que una formulación elegante del descenso en la dirección del gradiente. Como ayuda a la búsqueda de soluciones adecuadas se ha optado por utilizar técnicas encaminadas a solventar este problema. En este trabajo se describe la utilización combinada de dos algoritmos de aprendizaje: retropropagación (*backpropagation*) y búsqueda aleatoria progresiva (*creeping-random-search*) para tratar de mejorar la capacidad predictiva del modelo. A pesar de la diferente naturaleza de cada una de las técnicas (determinista frente a aleatoria) se ha visto que su utilización conjunta^[31] es adecuada ya que, la retropropagación tiende a ser más rápida en convergencia aunque cae a menudo en mínimos locales, mientras que la búsqueda aleatoria progresiva converge en general más despacio, pero tiene un índice de éxito mayor en cuanto a evitar mínimos locales. La combinación de esta búsqueda aleatoria con el aprendizaje determinista se puede implementar desde diferentes puntos de vista. A parte de la lógica generación aleatoria de pesos y sesgos iniciales, uno de los más usados es el proceso denominado recocido simulado^[32], el cual trata de buscar aleatoriamente soluciones mejores en las vecindades de la solución alcanzada hasta el momento. Sin embargo, se ha observado^[33] una mejora en convergencia del 60 % al 85 % de los casos si los pesos y sesgos se perturban con una búsqueda aleatoria en el momento de aprendizaje de la retropropagación (en vez de saltar a otra solución), esto es, combinando una búsqueda aleatoria progresiva con la retropropagación.

Así pues, la inclusión de la búsqueda aleatoria progresiva en la red neuronal se basa en una búsqueda a través de procesos aleatorios, de forma que a medida que el aprendizaje con retropropagación va convergiendo a una solución, nuevas combinaciones aleatorias de pesos buscan mejores soluciones que de otra manera quedarían ocultas para la retropropagación. Para que el método tenga coherencia, esta aleatoriedad no se usa en todas las direcciones, sino que dicha búsqueda aleatoria trata de ayudar y complementar a

la retropropagación. El método es sencillo y se introduce bien en cualquier código base de perceptrón multicapa. Esta técnica se ha usado para crear un modelo de redes neuronales que sea capaz de predecir la cantidad de austenita retenida en función de los parámetros de procesamiento de aceros TRIP de bajo silicio y/o aluminio. Los parámetros seleccionados representan los principales factores que afectan a la cantidad de austenita retenida durante el procesamiento del acero. Dichos parámetros son carbono, manganeso, silicio, aluminio (en cuanto a composición química), T_1 y t_1 (temperatura y tiempo de tratamiento térmico intercrítico) y T_2 y t_2 (temperatura y tiempo de tratamiento térmico bainítico). La base de datos cuenta con 352 datos recopilados de la bibliografía^[4-30].

2. RETROPROPAGACIÓN Y BÚSQUEDA ALEATORIA PROGRESIVA

El proceso de aprendizaje clásico de la retropropagación en una red neuronal sigue una serie de pasos que modifican los pesos que definen la red neuronal usando el error final cometido como objetivo a minimizar. Una observación de los pasos de este algoritmo nos muestra cómo y donde actúa la búsqueda aleatoria progresiva.

De forma esquemática, en cada paso del algoritmo de la retropropagación se han de calcular iterativamente las variaciones (s^m) de cada peso, usando la variación anterior para modificar éstos. Debido a que el objetivo es el de minimizar el error cuadrático de la salida de la red con los datos de referencia (t), la secuencia de estas funciones queda como sigue:

$$\begin{aligned} s^M &= -2F^M(n^M)(t - a) \\ s^m &= F(n^m)(W^{m+1})'s^{m+1}, \quad \forall m = M-1, \dots, 2, 1 \end{aligned} \quad (1)$$

Dónde F es la matriz definida como

$$F^m(n^m) = \begin{pmatrix} f^m(n_1^m) & 0 & \dots & 0 \\ 0 & f^m(n_2^m) & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & f^m(n_{s^m}^m) \end{pmatrix} \quad (2)$$

Con f^m la función de la capa m -ésima y

$$n_i^m = \sum_{j=1}^{s^{m-1}} (w_{i,j}^m a_j^{m-1} + b_i^m) \quad (3)$$

en el que, a^{m-1} son las salidas de las neuronas de la capa anterior. Finalmente se actualiza la matriz de pesos (W) y sesgos (b)

$$\begin{aligned} W^m(k+1) &= W^m(k) - \alpha s^m (a^{m-1})' \\ b^m(k+1) &= b^m(k) - \alpha s^m \end{aligned} \quad (4)$$

dónde α es el llamado índice de aprendizaje.

Si la red tiene varias capas, las variaciones s^m pueden adoptar valores ínfimos, debido a que se obtienen de la derivación de funciones con variación casi nula en gran parte de su dominio. El algoritmo de la retropropagación calcula estas variaciones iterativamente usando su predecesora, por lo que si una de ellas roza valores nulos, éstos se propagan sucesivamente, haciendo que las variaciones en los pesos asociados a estas zonas de la red neuronal sean pequeñas, promoviendo más variación en pesos que no se vean afectados.

A modo de descripción visual, la figura 1 muestra cómo una de las neuronas (ϵ) de la última capa (en blanco) ha obtenido valores de n^M que al ser evaluada por la función-matriz F^M darán como resultado valores casi nulos en componentes de s^M . En primer lugar, los pesos de esta neurona sufrirán una variación próxima a cero ($W^M(k+1) \approx W^M(k)$) en las componentes nulas de s^M y provocarán que los pesos de las neuronas cuyos valores de salida sean usados después en dicha neurona no tengan variaciones significativas, por ser éstas (s^{M-1}) calculadas usando la función s^M anterior, con componentes de valor casi nulo. Esta segunda capa de neuronas, aunque no se verá totalmente afectada por la neurona ϵ anterior, heredará en los pesos asociados a esta la neurona una disminución en el orden de magnitud de la variación en sus pesos. La información contenida en la matriz $F(n^{M-1})$, destinada a describir la variación de los pesos usados como entrada de la neurona de la siguiente capa desaparece. El descenso en el orden de magnitud de la variación se traspasa a la siguiente capa, diluyéndose poco a poco.

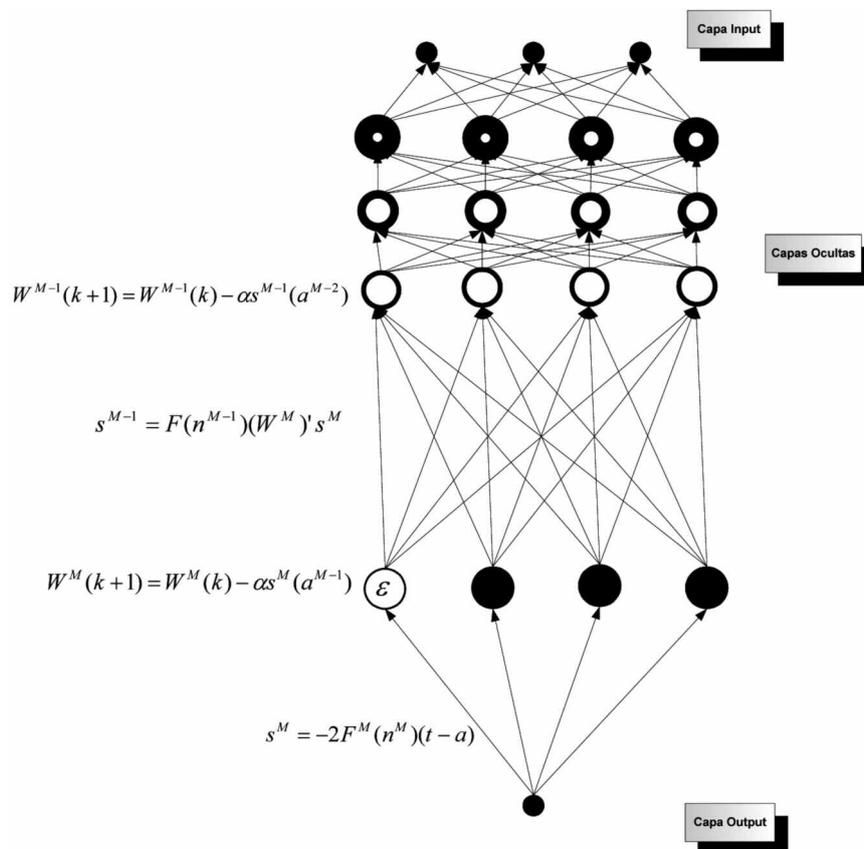


Figura 1. Propagación de las derivadas nulas en una red neuronal. La variación nula de una neurona afecta a sus predecesoras.

Figure 1. Propagation of the zero derivatives in a neural network. The zero variation of a neuron affects its predecessors.

En cada paso del algoritmo de la retropropagación, los pesos de cada neurona afectada variarán ínfimamente, y con el tiempo, obligará a los pesos del resto de neuronas de su capa a modificarse para minimizar el error con respecto a un valor de su peso asociado prácticamente inamovible. Esto es lo que produce que los mínimos más cercanos a estos valores iniciales atraigan la solución sin opción a saltar a soluciones con menor error, debido a que al no modificarse estos pesos, el resto debe adecuarse para minimizar el error en la medida de lo posible, cayendo en un mínimo local.

Se podría deducir que una selección cuidadosa de los pesos iniciales pudiera salvar este problema, pero es fácil caer en estas combinaciones de pesos que provocan ralentización en el aprendizaje. Los valores de la derivada de la tangente hiperbólica en un rango de evaluación normal pueden ser varios órdenes de magnitud inferiores a 1 con facilidad. Aunque la normalización de los datos contribuiría a evitar este problema, un número alto de entradas contribuye a alcanzar valores problemáticos.

Por lo tanto, los perceptrones multicapa con muchas capas pueden sufrir de la aparición de mínimos locales, con cierta facilidad para caer en ellos. Así, la búsqueda inicial de funciones que engloben de manera directa las propiedades a estudiar es fundamental para poder definir perceptrones multicapa de una o dos capas y el menor número posible de neuronas. Incluso en redes neuronales simples existe el riesgo de caer en zonas de lenta variación. Por tanto, se ha combinado el tradicional retropropagación con la búsqueda aleatoria progresiva para incrementar las variaciones de estos pesos problemáticos.

Aunque sólo una capa de neuronas se usa para este modelo, el fenómeno de la lentitud en la convergencia de los pesos de zonas con derivadas casi nulas se puede presentar también (aunque en menor medida que con varias capas). El problema no se propagará de capa en capa, pero afectará a los pesos de la misma capa al producir una tendencia hacia mínimos que suplan la carencia de variación de los pesos que definen la función de evaluación de una neurona.

Así pues, la búsqueda aleatoria progresiva es un método de búsqueda basado en procesos aleatorios, de forma que, a medida que se va convergiendo a una solución, nuevas combinaciones aleatorias de pesos buscan mejores soluciones a las ofrecidas por el algoritmo de la retropropagación (o soluciones fuera de mínimos locales). Esta aleatoriedad no se usa en todas las direcciones, sino que, en el caso de búsqueda aleatoria progresiva se intenta ayudar a la retropropagación cuando éste algoritmo ofrece

variaciones ínfimas en sus funciones sin que esto corresponda a una solución óptima. El método es sencillo y se introduce bien en cualquier código base de perceptrón multicapa.

Del cálculo de las derivadas del algoritmo de la retropropagación se obtiene la dirección de máxima variación (s_{max}) de cada peso (Ecs. 1) y 4)). El método de búsqueda aleatoria progresiva busca aleatoriamente soluciones mejores por medio del descenso en la dirección del gradiente y aprovecha la dirección de máxima variación de los pesos para buscar posibles soluciones dentro de un cono, alrededor de esta dirección de máxima variación (vector s_{max}) y con pasos ligeramente más grandes para poder salir de los mínimos locales.

En aquellos pesos que no sufran estas pequeñas variaciones, la búsqueda aleatoria progresiva no cambiará las soluciones, puesto que una pequeña variación en la dirección de búsqueda, cuando las derivadas no son pequeñas, no afecta a la solución final, pero sí ofrece la posibilidad de cambiar aquellos pesos que sí la tienen, como se muestra en la figura 2. En esta figura se representa el vector de máxima variación en un paso (s_{max}). Su proyección en el eje Y ($s_{max(y)}$) corresponde a la variación de un peso al que se le permite variar con libertad con retropropagación. Se comprueba que la diferencia entre un método y otro no afecta su variación y tenderá sin modificaciones hacia donde el algoritmo le obligue a variar. En el eje "X", se representa un caso hipotético de variación sobre una zona de derivada nula ($s_{max(x)}$). Aquí la búsqueda aleatoria progresiva permite explorar zonas prohibidas para este peso. En el caso en el que no se encuentren soluciones mejores, el siguiente peso será similar, y el método no variará la solución de retropropagación. Pero si existen soluciones mejores, el método ofrece la posibilidad de salir de estos mínimos locales. Por tanto el método de la búsqueda aleatoria progresiva es coherente con el retropropagación y ambos métodos se complementan bien, sin perder soluciones.

3. CREACIÓN DE UN MODELO DE REDES NEURONALES PARA LA PREDICCIÓN DE AUSTENITA RETENIDA EN ACEROS TRIP

3.1. Base de datos

En este trabajo se determinará la cantidad de austenita retenida en función de los diferentes parámetros del procesado del acero, que incluye una ruta térmica clásica como se muestra en la figura 3. Se parte de un acero con ferrita y perlita que se transforma parcialmente

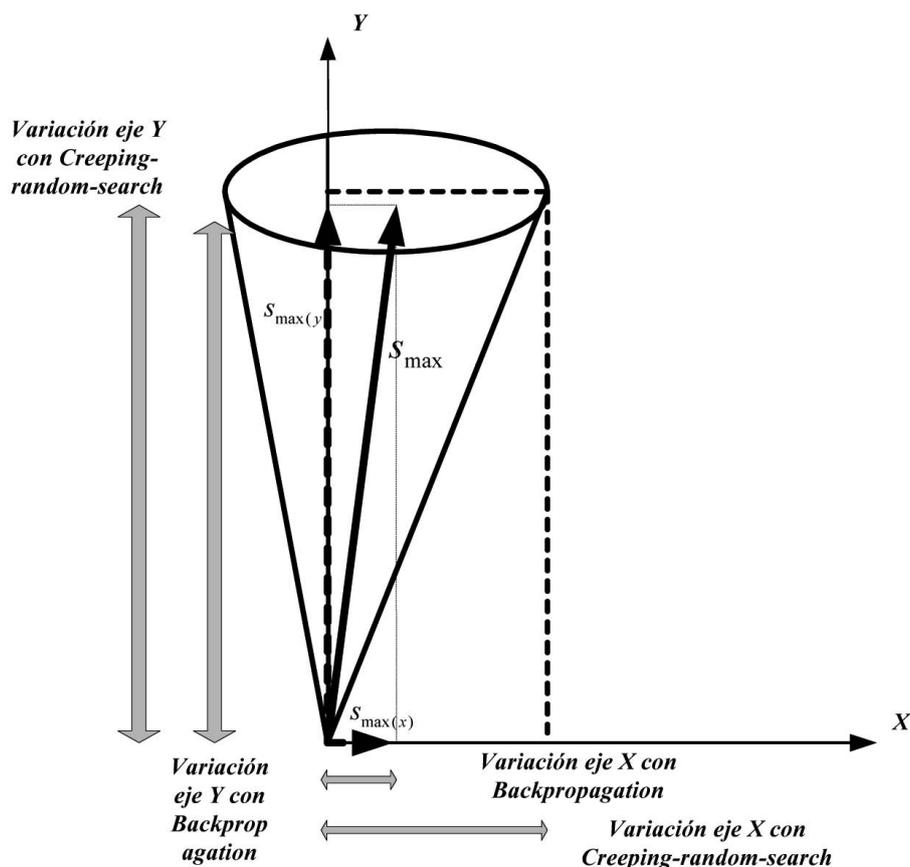


Figura 2. Variación de los pesos usando retropropagación y búsqueda aleatoria progresiva.

Figure 2. Weight variation using backpropagation and creeping-random-search.

en austenita en el primer tramo (a temperatura T_1 y tiempo t_1), que se sitúa entre las temperaturas mínima A_{c1} y máxima A_{c3} para este proceso. A continuación, se enfría el acero a velocidades altas (normalmente 50 K/s) y se aplica un segundo tratamiento isotérmico (T_2 , t_2). En este tramo parte de la austenita se transformará en bainita, mientras que la aleación con manganeso permite evitar la formación de fases no deseadas. Por último se temple el acero hasta temperatura ambiente y sólo la austenita con altas concentraciones de carbono y aluminio y/o silicio se retendrá a temperatura ambiente. La austenita que no cumpla estas condiciones se transformará a martensita. Por lo tanto, el modelo desarrollado tiene que predecir la cantidad de austenita retenida en función de los principales parámetros de composición química (C, Mn, Si y Al) y del tratamiento térmico (T_1 , t_1 , T_2 , t_2).

Se ha creado una base de datos con 352 elementos a partir de los valores experimentales encontrados en la literatura^[4-30]. En la tabla I se muestran las 8 variables

de entrada utilizadas para crear el modelo, así como el promedio y la desviación típica de cada variable.

3.2. Elección de las funciones y descripción de la red neuronal

Históricamente, el uso de tangentes hiperbólicas (o funciones sinusoidales) como funciones de las neuronas ha sido muy común en la metalurgia, debido a que en general las propiedades estudiadas sugieren una tendencia como las que estas funciones desarrollan. Además, la combinación y composición de esta clase de funciones ofrecen una gran diversidad en el resultado final, lo que permite minimizar el error de casi cualquier base de datos, aunque requiera un gran número de neuronas y capas. No obstante un análisis de los datos sugiere el uso de otro tipo de funciones más adecuadas, de forma que se reduzca el número de neuronas necesarias, y así converger más

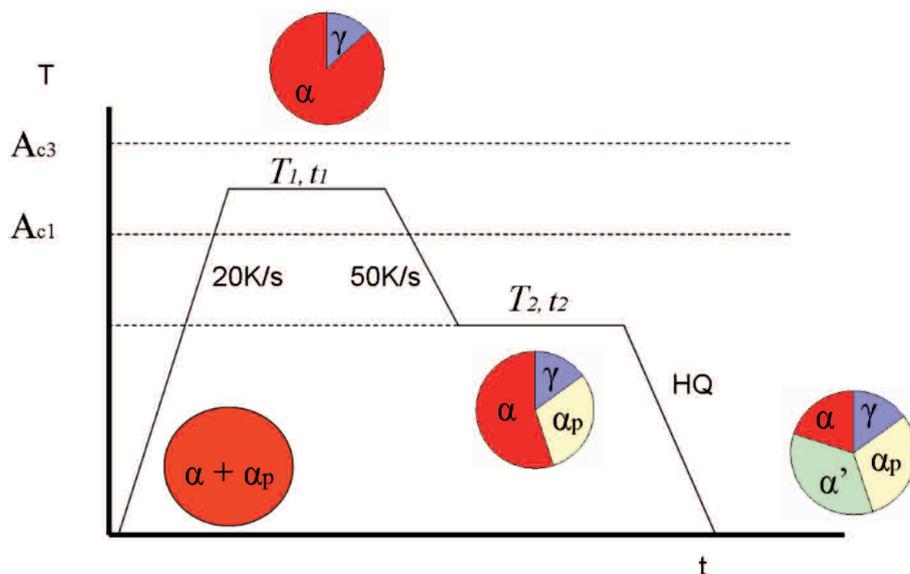


Figura 3. Ruta para la producción de aceros multifase. α : ferrita; α_p : perlita; γ : austenita; α_b : bainita; α' : martensita; γ_r : austenita retenida; HQ: temple con helio.

Figure 3. Route for the production of multiphase steels. α : ferrite; α_p : pearlite; γ : austenite; α_b : bainite; α' : martensite; γ_r : retained austenite; HQ: helium quench.

Tabla I. Variables de entrada para la red neuronal, con el valor promedio y la desviación estándar correspondientes a los 352 aceros. Las concentraciones están en %-peso, las temperaturas en °C y los tiempos en segundos

Table I. Input variables for the neural network with their corresponding mean value and standard deviation of the 352 steels. Contents are in weight percent, temperatures in °C and time intervals in seconds

Elemento	Intervalo	Promedio	Desv. Típica
C	0,07 - 0,47	0,20	0,09
Mn	1,2 - 1,8	1,49	0,14
Si	0 - 2,0	0,77	0,59
Al	0 - 2,0	0,61	0,65
T_1	730 - 1.040	809	49
t_1	50 - 1.800	351	344
T_2	310 - 500	412	38
t_2	0 - 36.000	674	2.412

rápida y a menos mínimos locales, saltando de manera más sencilla al mínimo absoluto. Además, el uso de sucesivas capas ralentiza la convergencia,

no sólo por el incremento de operaciones, si no por el fenómeno variaciones ínfimas descrito anteriormente, y que trata de ser mejorado con la combinación de ambos métodos de aprendizaje.

Del análisis de la base de datos recopilada se observa que el aumento del tiempo de calentamiento t_2 a la temperatura T_2 produce un aumento de la austenita retenida, hasta que alcanza su valor máximo para una determinada composición química. Una vez alcanzado el máximo descende la cantidad de austenita retenida suavemente hasta que desaparece por completo. Así pues, se hace necesario el uso de funciones capaces de describir este comportamiento de una manera simple. Esto redundará en la sencillez y facilidad del algoritmo de aprendizaje para encontrar los mínimos que describen la solución deseada. La función descrita en la ecuación (5) describe la tendencia descrita.

$$f(g(x) = e^{-g(x)}, g(x) = \left(\sum_{j=1}^{s^{m-1}} w_{i,j}^m x + w_{i,j}^m \sqrt{x} + b_i \right)^2 \quad (5)$$

Esta función de fácil manejo, derivable y bien definida para valores positivos está muy bien condicionada para el uso dentro de una red neuronal, puesto que los pesos y sesgos se usan aquí en forma de

combinación lineal. Además teniendo en cuenta que para valores nulos de t_2 existe austenita retenida (es decir, su valor no es 0 en el origen), estas funciones ofrecen claras ventajas de convergencia. Una representación visual de esta familia de funciones puede verse en la figura 4 para varios valores de los pesos.

No obstante, las funciones cuadráticas exponenciales son muy sensibles a caer en zonas con valor nulo en su derivada, por lo tanto, la combinación de los algoritmos retropropagación y la búsqueda aleatoria progresiva se ha usado para prevenir los mínimos locales.

La estructura de la red neuronal viene descrita en la figura 5. Ocho datos de entrada (cuatro de composición, dos de tratamiento térmico intercrítico y dos de tratamiento térmico bainítico) dos capas principales de asignación de pesos y evaluación de las funciones y la capa de salida con una única salida que corresponde al porcentaje de austenita retenida.

3.3. Desactivación de neuronas

Una vez definida la red neuronal, sus sistemas de aprendizaje y el modo de procesar la información de las neuronas (sus funciones), se ha aplicado este conocimiento para sintetizar las propiedades de la austenita retenida. En este proceso, se sobrepasa frecuentemente el número necesario de neuronas, incrementando de esta manera el problema llamado de sobre ajuste.

Una vez creado un modelo aceptable se aplica una desactivación paulatina de las neuronas que menos afecten al modelo. Las neuronas a desactivar se seleccionan en función de su influencia en el modelo. Se llevan a cabo dos tipos de desactivación: desactivación de pesos por baja influencia en el modelo, y desactivación de neuronas por hiperplanos paralelos.

3.4. Desactivación de pesos por baja influencia

Dentro de una neurona, aquellos pesos que no sobrepasen un umbral de activación, se considerarán nulos. Este umbral de activación, detectado en las neuronas biológicas, se usa aquí para comprimir la información en el mínimo número posible de neuronas. De esta manera, al ejecutar el algoritmo aquellos que se mantienen activos recogen la información contenida anteriormente en los pesos desactivados y así la red se comprime. A medida que se converge a soluciones que mantienen la tendencia de los datos se va disminuyendo el número de pesos. Esto no desactivará una neurona, sino parte de sus pesos. Estos pesos, que con anterioridad generaban menores errores, daban información local de los datos, pero disminuyen las posibilidades de extrapolar a zonas donde no hay datos, ya que esos pesos no contienen información a nivel global, no así como los pesos que sí describen globalmente el comportamiento.

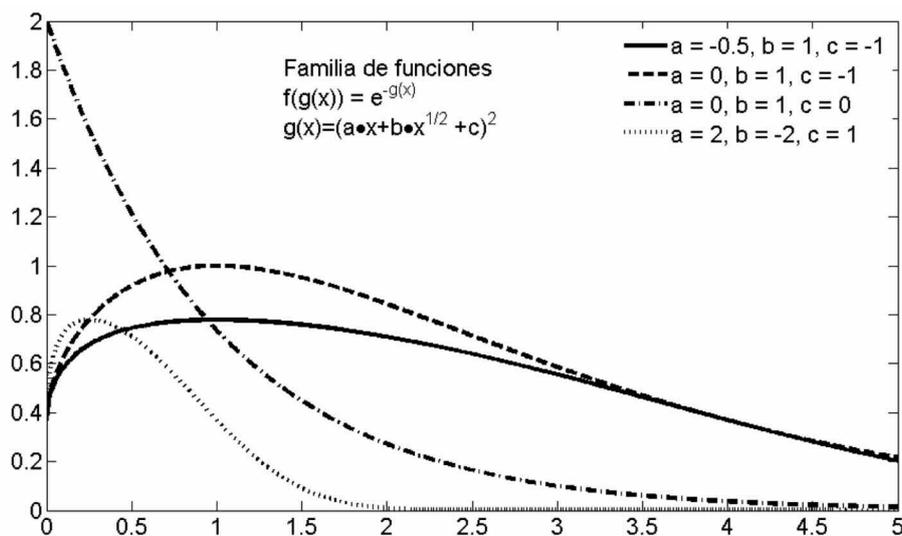


Figura 4. Representación de la familia de funciones elegida en la red neuronal para varios valores de los pesos.

Figure 4. Representation of the function family for the neural network with different weight combinations.

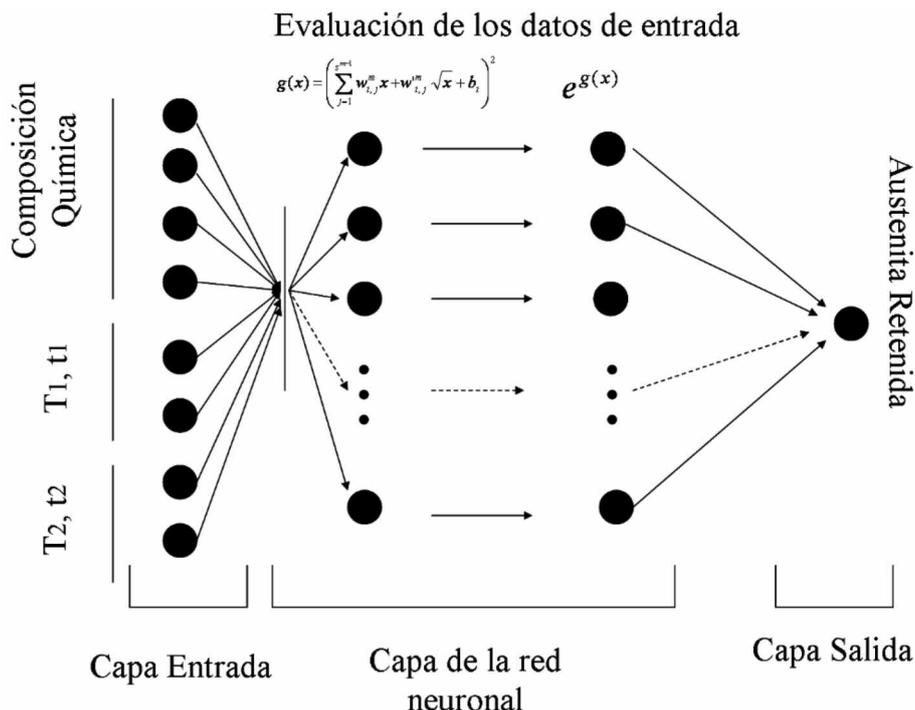


Figura 5. Definición de la estructura de la red neuronal. Capa de entrada, evaluación y salida.

Figure 5. Neural network structure. Input layer, computation and output.

En los modelos creados, se ha podido llegar a desactivar más del 50 % de los pesos, sin una pérdida significativa en el cálculo del error en la salida de la red con respecto al valor deseado. La desactivación de un peso de la segunda capa, provoca automáticamente la desactivación de todos los pesos de la capa anterior que derivaban en esa neurona. Este hecho reduce en mayor medida una red neuronal con varias capas, ya que si hay m neuronas por capa, al anular un peso de la última capa, anularía m pesos de la anterior, y éstos a la vez anularían a m^2 pesos de su predecesora.

3.5. Desactivación de neuronas por hiperplanos paralelos

En los modelos con muchas neuronas en una capa, se puede producir el caso en el que dos neuronas de la misma capa multipliquen funciones iguales evaluadas en hiperplanos cercanos entre sí. Es decir, si la salida de una capa es

$$W_1^m f(n_1^m) + W_2^m f(n_2^m) + \dots + W_n^m f(n_n^m) + b^m \quad (6)$$

y se tiene que dos hiperplanos (n_i, n_j) son similares

$$n_i^m = \sum W_{i,j}^{m-1} a_j^{m-1} + b_i^{m-1} \approx \sum W_{k,j}^{m-1} a_j^{m-1} + b_k^{m-1} = n_k^m \quad (7)$$

$$f(n_i^m) \approx f(n_k^m)$$

Entonces, el peso W_k^m pasa a formar parte del peso W_i^m y se evalúa la función en hiperplano compuesto.

$$W_i^2 (f(n_i) + W_k^2 f(n_k)) \approx (W_i^2 + W_k^2) f\left(\frac{n_i + n_k}{2}\right) \quad (8)$$

A continuación se actualizan los pesos para conservar el error en la medida de lo posible.

3.6. Análisis del modelo creado

El sistema de la desactivación de las neuronas produce sucesivos modelos, desde el más complejo (24 neuronas por capa) hasta el más simple posible sin aumento significativo del error (14 neuronas por capa). En general, cuanto más complejo sea, mejor será el ajuste de los datos, y a medida que se simplifica el modelo, el error aumenta, especialmente en los estadios finales de la simplificación. Los primeros pa-

sos de este proceso pueden dar como resultado modelos con menos neuronas e igual error. Esto hecho es lógico, puesto que cuando se trabaja con un orden de magnitud de 20-30 neuronas por capa, si las menos significativas son desactivadas, es razonable que la evolución que éstas ayudaban a mostrar en el modelo sea recogida ahora por otras (o combinaciones de otras) al ejecutar el algoritmo con esta nueva disposición de los pesos. Este hecho se producirá en más medida cuando un peso de la segunda capa se desactiva y su hiperplano correspondiente tenga uno muy similar en una neurona no desactivada, debido a que en realidad, la información está duplicada, y solo hay que combinarla.

De esta manera, se pueden usar los diferentes modelos dependiendo del objetivo deseado. Para interpolaciones cercanas a puntos de la base de datos inicial puede ser preferible el uso de modelos más complejos, que ajusten mejor los datos, pero a medida que el modelo se aleja de los datos iniciales, es preferible usar modelos más simples, no tanto para tratar de obtener datos precisos sino para estudiar la tendencia de la concentración de austenita retenida. La tabla II muestra la evolución media del error (del tanto por ciento de concentración de austenita retenida) y su desviación típica a medida que se desactivan pesos, partiendo de un modelo de 24 neuronas.

Con estos modelos, el 60 % de los datos están descritos con un error inferior a un 1 % en su composición final y un 86 % ya estaría por debajo de un 2 %. Debido a que los elementos de la composición química de un acero menos significativos (como el fósforo, azufre, niobio, nitrógeno, titanio y cromo) no han sido incluidos a la hora de desarrollar el modelo, ni la velocidad de enfriamiento entre T_1 y T_2 , los errores serán mayores, puesto que el modelo simplificado no refleja toda la evolución de la formación de austenita retenida. Así pues, el modelo final comprende 14 neuronas por capa con un error similar en media y desviación típica al de 24 neuronas una vez aplicado ambas técnicas de desactivación de neuronas.

4. VALIDACIÓN Y APLICACIÓN DEL MODELO CREADO: DETERMINACIÓN DE LA AUSTENITA RETENIDA EN ACEROS MULTIFASE DE BAJO CONTENIDO EN ALUMINIO Y/O SILICIO

Para valorar el modelo creado se procede en dos etapas. Primero se comprueba el ajuste con los resultados de la base de datos^[4-30] cuyo error de manera general se presenta en la tabla II. En particular, para el acero TRIP de composición C = 0,24, Mn = 1,69, Si = 0,3, Al = 1,34, $T_1 = 810$ °C, $t_1 = 600$ s y $T_2 = 480$ °C^[7] se presenta en la figura 6 la evolución con la tiempo t_2 que el modelo refleja para una composición específica de un acero y se compara con las predicciones realizada por un modelo de redes neuronales donde sólo se ha utilizado el algoritmo de la retropropagación.

El acero considerado para la segunda etapa de validación ha sido tratado y analizado para este propósito y es por tanto ajeno a la base de datos con la que se crea la red neuronal. Tiene la composición química mostrada en la tabla III que ha tenido un mantenimiento intercrítico de $T_1 = 730$ °C con un tiempo $t_1 = 300$ s y una temperatura de mantenimiento bainítico de 370 °C. La cantidad de austenita retenida se determina por medio de un proceso de ataques químicos sucesivos con el reactivo Nital al 2 % (10 s) seguido de un ataque con una solución acuosa de metabisulfito sódico al 30 % (30 s). La imagen filtrada con luz azul muestra como se revela la austenita retenida, y se diferencia de la martensita y bainita de la microestructura.

Sin embargo, es necesario corroborar que este proceso de ataque y observación con luz azul revela efectivamente la austenita retenida. En este sentido se ha procedido a comprobar por medio de difracción de rayos X (XRD)^[34] los valores de austenita retenida en distintas microestructuras (las medidas se realizaron en un difractor D-5000 aplicando el

Tabla II. Evolución del modelo mediante la desactivación de pesos

Table II. Evolution of the model by means of weight deactivation

nº pesos $W_2 = 0$	nº pesos $W_1 = 0$	% pesos desactivados	Media error cuadrático	Des. Típica error
5	218	36,5 %	0,0758610	1,0012
6	250	40,4 %	0,0758597	1,0023
7	268	43,2 %	0,0758575	1,0023
9	313	53,0 %	0,0847520	1,1017
10	330	56,0 %	0,0898370	1,0928

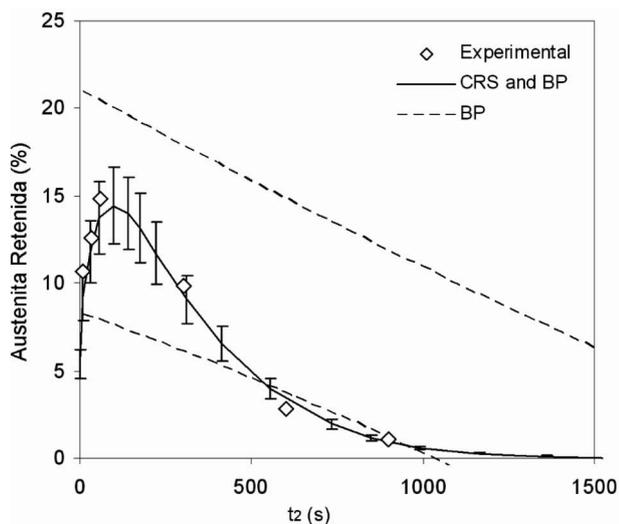


Figura 6. Comparación del modelo con datos reales de la base de datos. Evolución con respecto a t_2 . ($C = 0,24$, $Mn = 1,69$, $Si = 0,3$, $Al = 1,34$, $T_1 = 810$ °C, $t_1 = 600$ s y $T_2 = 480$ °C). BP indica las predicciones del modelo sólo con retropropagación (límites indicados por la línea de puntos) y CRS+BP las del modelo combinado con barras de error relativo a la desviación típica del error.

Figure 6. Comparison between the predicted evolution and the real data from data base as a function of t_2 ($C = 0.24$, $Mn = 1.69$, $Si = 0.3$, $Al = 1.34$, $T_1 = 810$ °C, $t_1 = 600$ s, and $T_2 = 480$ °C). BP stands for back-propagation model and CRS for creeping random search algorithms, respectively with error bars related with standard deviation error. Dashed lines indicate the limits for BP model.

método de reflexión de Schultz). Una comparación de los resultados obtenidos con ambas técnicas nos permite concluir que la técnica de ataque químico tiene un error inferior al 1 % en austenita retenida observada por lo que se revela como una técnica efectiva y fácil de aplicar en estos aceros multifase.

A partir de la determinación de la fracción de austenita retenida en micrografías, se puede determinar experimentalmente la evolución de la austenita retenida el tiempo (t_2) de mantenimiento en la zona bainítica del acero de la tabla III. De esta forma, se puede comprobar la evolución predicha por el modelo con la evolución real obtenida experimentalmente. Los resultados de dicha comprobación se muestran en la figura 7. Se comprueba que existe un buen ajuste de la evolución del contenido de austenita retenida, y que el modelo cumple los objetivos iniciales de la investigación.

Tabla III. Composición química del acero de validación (% en peso)

Table III. Composition of the studied steel (in wt.-%)

C	Mn	Si	P	S	Cr	Al	Cu
0,20	1,1	0,34	0,010	0,011	0,011	0,036	0,018

Una vez validado el modelo, éste puede utilizarse también para determinar los mapas de transformación de un acero multifase. A medida que t_2 aumenta, la formación de bainita induce un enriquecimiento en carbono de la austenita que no ha transformado aun, lo que lleva a una estabilización térmica de la misma. Este proceso se acelera para el caso de temperaturas T_2 altas ya que para tiempos largos el revenido de la bainita empieza a inducir un proceso de precipitación que desestabiliza al austenita. Sin embargo, para valores bajos de T_2 , la ralentización en la cinética de formación de bainita

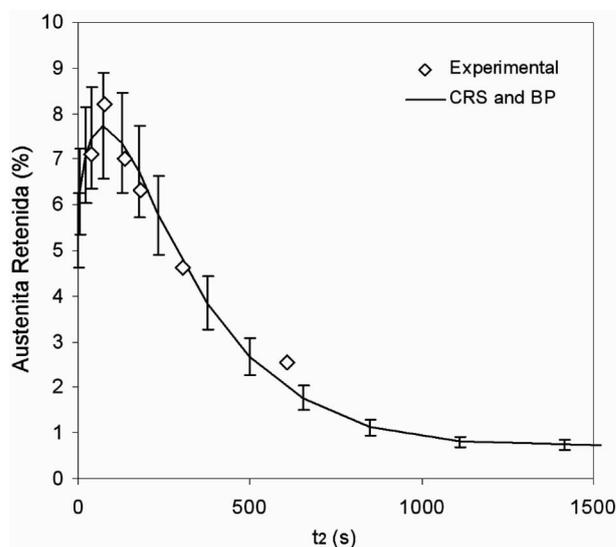


Figure 7. Comparación entre los resultados experimentales y predichos por el modelo para la determinación de austenita retenida del acero de la tabla III con, $T_1 = 730$ °, $t_1 = 300$ s, y $T_2 = 370$ °C con barras de error relativo a la desviación típica del error.

Figure 7. Comparison between predicted and experimentally determined retained austenite volume fraction in steel of table III with, $T_1 = 730$ °, $t_1 = 300$ s, y $T_2 = 370$ °C with error bars related with standard deviation error.

provoca que el enriquecimiento en carbono necesario para estabilizar la austenita no sea suficiente y ésta transforma casi completamente a martensita. La austenita se estabiliza gradualmente para valores de t_2 mayores. Dada la complejidad en la transformación de fases a lo largo del proceso, una optimización en los parámetros de tratamiento para un acero dado (temperatura y tiempo intercrítico, y temperatura y tiempo de mantenimiento bainítico) puede mejorar las propiedades del acero obteniendo la cantidad deseada de austenita retenida.

5. CONCLUSIONES

- La combinación de los algoritmos de la retropropagación y búsqueda aleatoria progresiva ha sido utilizada para la creación de una red neuronal que puede predecir la cantidad de austenita retenida final en aceros TRIP. El objetivo de esta combinación es complementar al aprendizaje clásico de la retropropagación y tratar de buscar soluciones mejores que las posibles solamente con dicho método de aprendizaje. La combinación de ambos métodos solo tiene efecto cuando el retropropagación sufre ralentización en su aprendizaje. Así pues, la búsqueda aleatoria progresiva es capaz de modificar el aprendizaje de la red neuronal para ayudar a encontrar soluciones de pesos que definan la red neuronal y optimicen del error.
- La desactivación de neuronas, por debajo de un umbral, ha sido fundamental en el proceso de simplificación del modelo, y ha producido una reducción superior al 50 % del tamaño de la red, sin un significativo aumento del error.
- La aplicación de este método combinado al estudio de la fracción de austenita retenida en los aceros TRIP ha dado lugar a una predicción fiable en función de los principales parámetros de entrada. No obstante, los errores del modelo en ciertos rangos de los datos sugieren la introducción de nuevos parámetros que ayuden a describir mejor el proceso de modelización de esta particular fase.

Agradecimientos

I. Toda Caraballo agradece al Ministerio de Ciencia e Innovación por su apoyo económico en forma de FPI. Los autores están muy agradecidos por el apoyo económico al Ministerio de Ciencia e Innovación por financiar esta investigación en el marco del Plan Nacional 2009 ENE2009 13766-C04-01.

REFERENCIAS

- [1] M.T. Hagan, H.B. Demuth y M. Beale, *Neural Network Design* Vol1. PWS Publishing Company, 1ª Ed., Boston, USA, 1996.
- [2] S.M.K. Hosseini, A. Zarei-Hanzaki, M.J. Yazdan Panah y S. Yuce, *Mater. Sci. Eng. A* 374 (2004) 122-128.
- [3] D.J.C. MacKay, *Neural. Comp.* 4(3) (1992) 448-472.
- [4] I.D. Choi, D.M. Bruce, S.J. Kim, C.G. Lee., S.H. Park y D.K. Matlock, *ISIJ Int.* Vol. 42 (2002) 1.483-1.489.
- [5] H.C. Chen, H. Era y M. Shimizu, *Metall. Trans.* 20A (1989) 437-445.
- [6] A. De Meyer, D. Vanderschueren y B.C. de Cooman, *ISIJ INT* Vol. 39 (1999) 813-822.
- [7] A. De Meyer, J. Mahieu y B.C. De Cooman, *Mat. Sci. Techn.* Vol. 18 (2002) 1.121-1.132.
- [8] E. Emadoddin, A. Akbarzadeh y Gh. Daneshia, *Mat. Char.* Vol. 57 (2006) 408-413.
- [9] N. Fonstein, N. Pottore, S.H. Lalam y D. Bhattacharya, *Mechanical Working and Steel Processing Conference*, Vol. 49, Ed. Association for Iron & Steel Technology, Warrendale, Chicago, USA (2003) pp. 549-561.
- [10] Q. Furnémont, M. Kempf, P.J. Jacques, M. Göken y F. Delannay, *Mat. Sci. Eng. A* Vol. 328 (2002) 26-32.
- [11] Q. Furnémont, *Tesis Doctoral*, Université Catholique de Louvain (2003).
- [12] S. Godet, *Tesis Doctoral*, Université Catholique de Louvain (2003).
- [13] T. Iung, J. Drillet, A. Couturier y C. Olier, *Proc. Conf. TRIP*, Vol.1, Gante, Bélgica, 2002, Ed. B.C. De Cooman, Steel Grips, Bad Harzburg, Germany, (2003) pp. 31-37.
- [14] P.J. Jacques, Q. Furnémont, F. Lani, T. Pardoen y F. Delannay, *Acta Mater.* Vol. 55 (2007) 3.681-3.693.
- [15] P.J. Jacques, E. Girault, Ph. Harlet y F. Delannay, *ISIJ Int.* 41 (2001) 1.061-1.069.
- [16] P.J. Jacques, E. Girault, A. Mertens, B. Verlinden, J. Van Humbeeck y F. Delannay, *ISIJ Int.* Vol. 41 (2001) 1.068-1.074.
- [17] S.J. Kim, C.G. Lee, T.H. Lee y C.S. Oh, *ISIJ Int.* Vol. 42 (2002) 1.452-1.456
- [18] D. Krizan, J. Antonissen, L. Barbé y B.C. De Cooman, *Mechanical Working and Steel Processing Conference*, Vol. 49, Ed. Association for Iron & Steel Technology, Warrendale, Chicago, USA (2003), pp. 437-448.
- [19] C.G. Lee, S.J. Kim, T.H. Lee y C.S. Oh, *ISIJ Int.* Vol. 44 (2004) 737-743.

- [20] C.G. Lee, S.J. Kim, C.S. Oh y T.H. Lee, *ISIJ Int.* Vol. 42 (2002) 1.162-1.168.
- [21] O. Matsumura, Y. Sakuma, Y. Ishii y J. Zhao, *ISIJ Int.* Vol. 32 (1992) 1.110-1.116.
- [22] A. Mertens, *Tesis Doctoral*, Université Catholique de Louvain (2002).
- [23] J. Ohlert, *Tesis Doctoral*, IEHK, RWTH Aachen, Shaker Verlag (2003).
- [24] B.S. Seonga, E.J. Shina, Y.S. Hana, C.H. Leea, Y.J. Kima y S.J. Kim, *Physica B* 350 (2004) 467-469.
- [25] W. Shi, L. Li, Y. Zhou, R.Y. Fu, X.C. Wei, B.C. De Cooman, P. Wollants, X.D. Zhu, y L. Wang, *Proc. Conf. TRIP*, Vol. 1, Gante, Bélgica, 2002 Ed. B.C. De Cooman, Steel Grips, Bad Harzburg, Germany, (2003) pp. 207-210.
- [26] A.K. Srivastava, D. Bhattacharjee, G. Jha, N. Gope y S. B. Singh, *Mat. Sc. Eng. A* 445-446 (2007) 549-557.
- [27] A.K. Srivastava, Y.K. Singh, G. Jha y N. Gope, *Mater. Char.* 57 (2006) 127-135.
- [28] S. Traint, A. Pichler, K. Spiradek-Hahn, K. Hulka y E. Werner, *Mechanical Working and Steel Processing Conference*, Vol. 49, Chicago, USA, 2003, pp. 579-594.
- [29] N.H. Van Dijk, A.M. Butt, L. Zhao, J. Sietsma, S.E. Offerman, J.P. Wright y S. Van der Zwaag, *Acta Mater.* 53 (2005) 5.439-5.447.
- [30] S. Zaefferer, J. Ohlert y W. Bleck, *Acta Mater.* Vol. 52 (2004) 2.765-2.778.
- [31] A. G. Korn, *Simulation*, Vol. 55 (4) (1990) 214-219.
- [32] A. Salvetti, B. Wilamowski, *ANNIE'94 Artificial Neural Networks in Engineering*, St. Louis, Missouri, USA, November 1994, pp. 13-16.
- [33] C.B. Owen y A.M. Abunawass, *Actas del World Congress on Neural Networks*, Vol. 1, New Jersey, 1993, pp. 625-630.
- [34] M. De Meyer, D. Vanderschueren, K. de Blauwe, B.C. De Cooman, *ISS, 41st Mechanical Working and Steel Processing Conference*, vol. XXXXVII. Baltimore, USA, 1999, Ed. Iron and Steel Society, 1999, pp. 483-491.