DESARROLLO DEL MODELO DE PREDICCIÓN DE LA TEMPERATURA DEL ARRABIO EN ACERÍA

Aida González García[*], Francisco Javier Fernández García[**], José Díaz Trapiella[**]

[*] Alumno; [**] Tutores

UO166144@uniovi.es

Departamento de Energía. Universidad de Oviedo.

RESUMEN

La predicción de la temperatura del arrabio líquido vertido en el convertidor de la acería en tiempo real proporciona una información útil en el proceso de conversión del arrabio en acero.

La medida de esta temperatura resulta complicada, conlleva incertidumbres y no se puede conocer en tiempo exacto.

En este trabajo se busca un modelo matemático que prediga la temperatura en tiempo real a partir de otras variables conocidas con la mayor exactitud posible mediante el estudio y creación de una red neuronal. La red neuronal será entrenada con una serie de datos de entrada y salida conocidos.

ABSTRACT

The prediction the liquid pig iron temperature loaded in the basic oxygen furnace of one steelmaking factory is very important. It gives us important information during the conversion of the pig iron into steel.

The measurement of this temperature is quite difficult. It has inherited some uncertainties and it is impossible to know at one exact time.

In this project we are looking for one mathematical model that allows us to predict the temperature at real time from other known measures by means of a neural network. The neural network will be trained with a series of known inputs-outputs.

1. INTRODUCCIÓN

ArcelorMittal Asturias es una planta integral siderúrgica que consta de dos factorías situadas en distintos concejos, unidas por una red ferroviaria de unos 25 km. Esta situación no es común sino que los altos hornos y las acerías suelen situarse próximas.

En la factoría de Gijón se encuentran dos hornos altos y en la de Avilés la acería LD.

El proceso siderúrgico comienza en los hornos altos. El horno alto trabaja de forma continua y periódicamente se extrae el arrabio líquido y la escoria. [1]

El arrabio líquido del horno alto se introduce en torpedos, Figura 1, que serán transportados desde los hornos altos hasta la acería LD en vagones. Cada torpedo puede tener arrabio procedente de un horno alto.

Figura 1. Vaciado de arrabio sobre carro torpedo [1]

El arrabio líquido sale del horno alto a unos 1530ºC (1803 K) y se carga en el torpedo previamente caliente (100-200ºC) para evitar el deterioro del revestimiento. En este tiempo de trasvase el arrabio pierde calor y se necesiten unos minutos para la estabilización de la temperatura.

El arrabio va disminuyendo su temperatura durante el transporte. [2, 3]

El torpedo pasa por un proceso de desulfuración, en el que se inyectan agentes desulfurantes.

En la acería el arrabio líquido de los hornos altos se transforma en acero.

Se vacía el arrabio de los torpedos por gravedad sobre unas cucharas situadas en un foso (Figura 2). Cada cuchara puede recibir arrabio procedente de dos o tres torpedos.

Figura 2. Trasvase de arrabio del torpedo a la cuchara

Una vez llena se saca con una grúa y se lleva a la estación de desescoriado. Tras este paso ya están preparadas para ir al convertidor.

El proceso de obtención del acero se lleva a cabo en el convertidor LD. El convertidor LD, Figura 3, es un recipiente basculante de acero revestido de material refractario que dispone de una lanza de inyección de oxígeno.

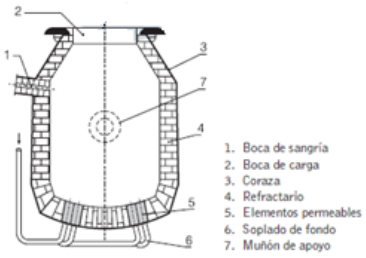

Figura 3. Convertidor LD

El convertidor se inclina 30º con respecto de la vertical y se carga con chatarra y restos de arrabio sólido, se vuelve a su posición vertical y se reparte homogéneamente el material al fondo. Se vuelve a inclinar 30º y se vierte el arrabio líquido de la cuchara. Ya en posición vertical se introduce una lanza de soplado de oxígeno puro a gran velocidad.

En la Figura 4 se observan las cargas de la chatarra y el arrabio en el convertidor.

Figura 4. Carga de la chatarra y el arrabio en el convertidor

Las reacciones de oxidación producidas en el convertidor son muy exotérmicas y dan lugar a temperaturas demasiado elevadas. Las temperaturas elevadas dan problemas con el refractario, deficiencias en la colada del acero líquido, desajustes en la composición química y mayores volúmenes de soplado de oxígeno.

La chatarra se utiliza como refrigerante en el proceso de conversión. El soplado de oxígeno, pureza mayor de 99.99%, se realiza para oxidar los elementos que acompañan al hierro del arrabio y se lleva a cabo a gran velocidad para asegurar la agitación y contacto de la colada. Además se cargan fundentes, ferroaleaciones, etc.

La temperatura a la que se encuentra el arrabio líquido en el momento del vertido al convertidor influye en la cantidad de chatarra que interesa cargar y el volumen de soplado de oxígeno. Esta temperatura no se puede medir en tiempo exacto e interesa predecir su valor para tener un conocimiento previo de la cantidad de chatarra que se añadirá al proceso de conversión.

En la actualidad la medida de la temperatura del arrabio líquido se realiza en la cuchara. La chatarra se introduce al convertidor antes que el arrabio líquido por lo que resultaría importante predecir esta temperatura con una buena precisión y conocer esta temperatura para preparar la carga de chatarra con antelación y el saber el volumen de oxígeno necesario.

En este estudio se busca un modelo predictivo. A partir de una serie de datos reales y mediante un aprendizaje el modelo prediga la temperatura del arrabio líquido, objeto del estudio. Para llegar a este modelo se trabaja con redes neuronales artificiales, las cuales ya han sido estudiadas en diversos artículos para variables y situaciones similares.

2. MÉTODO TRABAJO

2.1. Redes neuronales artificiales

Las redes neuronales artificiales son herramientas de búsqueda de modelos matemáticos utilizados en sistemas que no presentan un modelo adecuado o conllevan un algoritmo complejo. [4]

Una red neuronal está compuesta por un conjunto de neuronas. Cada neurona, Figura 5, recibe una serie de entradas a través de interconexiones caracterizadas por su peso, w, tiene asociado un sesgo, b, y una función de transferencia y emite una salida.

$$N_j = \sum x_i \cdot w_{ij} + b_j$$

$$y_j = f(N_j)$$

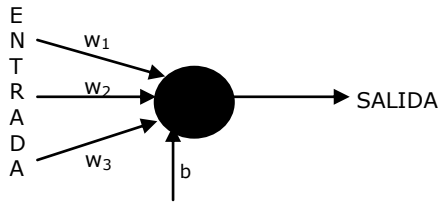

Figura 5. Neurona

La arquitectura de una red neuronal es la estructura de las neuronas y sus conexiones y la función de transferencia. La arquitectura está formada por capas de neuronas.

En la Figura 6 se representa una red neuronal formada por tres capas:

- Capa de entrada: Las neuronas reciben los datos de entrada procedentes del sistema de estudio.
- Capa de salida: Las neuronas emiten la respuesta de la red neuronal.
- Capa oculta: Proporciona grados de libertad a la red neuronal. La red puede estar formado por varias capas ocultas.

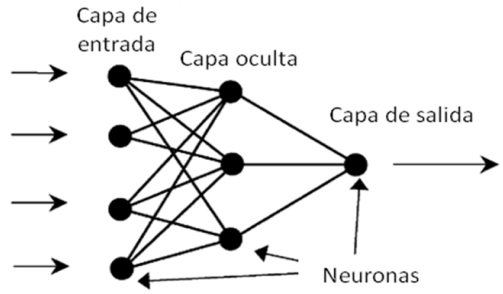

Figura 6. Red neuronal multicapa [5]

Según el flujo de las conexiones se puede distinguir entre redes unidireccionales y redes recurrentes o realimentadas.

A la red neuronal se le proporcionan unas variables de entrada conocidas y una variable de salida conocida que sería la variable objetivo o esperada. En este estudio la variable de salida es la temperatura del arrabio en la cuchara y las variables de proceso se van observando en distintas pruebas.

La búsqueda del modelo matemático consta de una serie de pasos en el trabajo con la red neuronal. En la Figura 7 se representan los distintos pasos que se llevan a cabo.

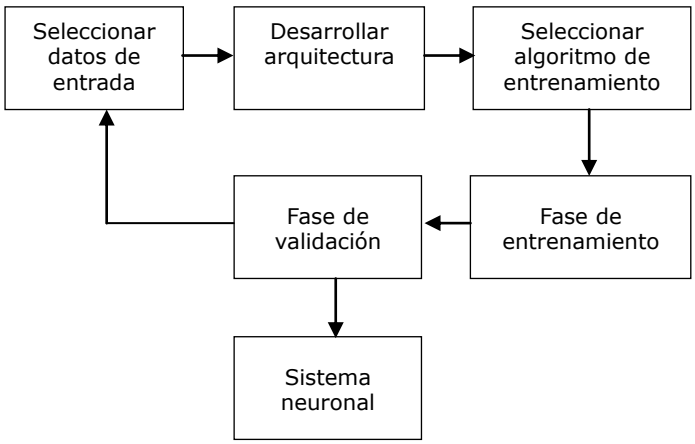

Figura 7. Diagrama de flujo de diseño de una red neuronal

El estudio de las redes neuronales se ha realizado con la herramienta de software matemático Matlab.

2.2. Procedimiento de las redes neuronales

En el trabajo realizado se han llevado a cabo una serie de redes neuronales modificando el número de variables de entrada, la arquitectura

(número de neuronas, función de transferencia) y algoritmo de entrenamiento.

- *Datos de entrada:* El número de variables del vector entrada debe ser suficiente para alcanzar un buen modelo pero esta selección puede resultar complicada puesto que la variable de salida puede no presentar relación con ellas o que algunas variables de entrada sean dependientes entre sí.
  Entre los datos conocidos se ha seleccionado entre 16 y 29 variables.
  Los datos originales se dividen en 3 procesos, entrenamiento (calcular gradiente de la función error y pesos actualizados), validación (parar el entrenamiento antes de que ocurra sobreajuste) y testado (predecir futuras ejecuciones y medir la calidad de la red).
- *Arquitectura:* Red unidireccional con una capa oculta en la que se varió el número de neuronas de la capa oculta, la conexión entre las capas, la función de transferencia. En la Tabla 1 se muestran las funciones de transferencia que se han utilizado.

Tabla 1. Funciones de transferencia [6]

Función	Relación entrada y salida		Matlab
Hard limit	a=0 a=1	n<0 n≥0	hardlim
Symmetrical hard limit	a=-1 a=1	n<0 n≥0	hardlims
Linear	a=n		purelin
Saturating linear	a=0 a=n a=1	n<0 0≤n≤1 n>1	satlin
Symmetrical saturating linear	a=-1 a=n a=1	n<-1 -1≤n≤1 n>1	satlins
Log-sigmoid	$a = \frac{1}{1+e^{-n}}$		logsig
Hyperbolic tangent sigmoid	$a = \frac{e^{n}-e^{-n}}{e^{n}+e^{-n}}$		tansig
Positive linear	a=0 a=n	n<0 n≥0	poslin
Competitive	a=1 a=0	Neuronas máx n Otras	compet

- *Algoritmo de entrenamiento:* Al buscar información sobre las redes neuronales se observó que en función del algoritmo de entrenamiento utilizado la duración de este proceso varía pero también la calidad del modelo. La Tabla 2 indica diferentes algoritmos de entrenamiento manejados.
- *Fase de entrenamiento:* Se lleva a cabo una inicialización de pesos y sesgos aleatoria, se siguen unos criterios de parada (número de iteraciones, tasa de aprendizaje, factor momento).
  En el aprendizaje se produce el ajuste de los pesos entre las conexiones de las capas. En cada neurona, j, con sus pesos y su función de transferencia.
- *Fase de análisis:* Tras el entrenamiento el test puede no ser adecuado porque la red alcance un mínimo local, no tenga suficientes neuronas para ajustar datos, esté sobreajustada, esté extrapolada.
  Se observan los errores de entrenamiento, validación y testado para deducir posibles mejoras y los valores de los pesos para saber si algunas variables no resultan importantes.

Tabla 2. Algoritmos de entrenamiento

Algoritmo	Matlab
Levenberg-Marquardt	trainlm
Bayesian Regularization	trainbr
BFGS Quasi-Newton	trainbfg
Resilient Backpropagation	trainrp
Scaled Conjugate Gradient	trainscg
Conjugate Gradient with Powell/Beale Restarts	traincgb
Fletcher-Powell Conjugate Gradient	traincgf
Polak-Ribiére Conjugate Gradient	traincgp
One Step Secant	trainoss
Variable Learning Rate Gradient Descent	traingdx
Gradient Descent with Momentum	traingdm
Gradient Descent	traingd

2.3. Desarrollo del modelo

El estudio que se va a realizar abarca las variables desde que el arrabio es introducido en los torpedos hasta que es introducido en el convertidor y el torpedo retorna. El proceso de estudio se refleja en la que muestra el paso del arrabio desde los altos hornos hasta el convertidor y del torpedo hasta que retorna.

En el trabajo se han tenido en cuenta posibles variables durante el proceso y se ha observado su efecto en el modelo.

Se ha considerado una red neuronal unidireccional de una capa de entrada, una capa oculta y una capa de salida.

Se ha ido variando el número de variables de entrada, el número de vectores de la capa de entrada, el número de neuronas de la capa oculta, la función de transferencia y el algoritmo de entrenamiento.

3. RESULTADOS Y DISCUSIÓN

3.1. Capa de entrada

Al comenzar a realizar el estudio se quiso observar el efecto de las variables de entrada, variando su orden y número, y el número de vectores de entrada sobre la red neuronal.

En primer lugar se utilizó la misma estructura:

- 3 capas.
- Funciones de transferencia: Hyperbolic tangent sigmoid y linear.
- Algoritmo de entrenamiento: Scaled Conjugate Gradient. Se utilizó este algoritmo de entrenamiento por ser el más rápido, aunque otros algoritmos nos proporcionaron mejores resultados en las siguientes pruebas.

La red neuronal utilizó como función de rendimiento el error cuadrático medio.

Al observar la Tabla 3 las pruebas primera y tercera tienen las mismas variables pero introducidas en distinto orden a la red neuronal, lo mismo pasa en las pruebas cuarta y quinta lo que refleja que el orden en que se introducen afecta a su rendimiento. Resulta más satisfactorio situar las variables por sus propiedades que por los torpedos.

También se dedujo que resultaba mejor un único vector de entrada. En el caso del número de variables de entrada no se llegó a una conclusión precisa

Tabla 3. Pruebas con los vectores y las variables de entrada

Prueba	Vectores de la capa de entrada	Variables	Regresión
Primera	1	16	0.38894
Segunda	4	5	0.14045
		5	
		5	
		1	
Tercera	1	16	0.070329
Cuarta	1	23	0.28229
Quinta	1	23	0.068866
Sexta	1	29	0.31476

Ciertos valores de las variables de entrada se introdujeron (sin modificar y modificados) para observar si se facilita el trabajo de la red pero la modificación disminuía su rendimiento. Además se realizó una eliminación de variables que pudieran aportar errores.

3.2. Capas ocultas y conexiones

Se realizaron ensayos con mayor número de capas ocultas y en el caso de pruebas con más de un vector una capa para cada vector pero este método no aportó mejora.

Se hicieron pruebas en las que la entrada se conectaba solo a la capa oculta, Figura 8, y en las que también se conectaba a la capa de salida, Figura 9.

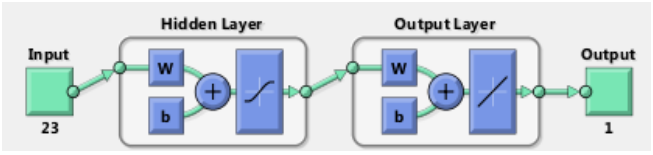

Figura 8. Red neuronal con conexión entrada con capa oculta [7]

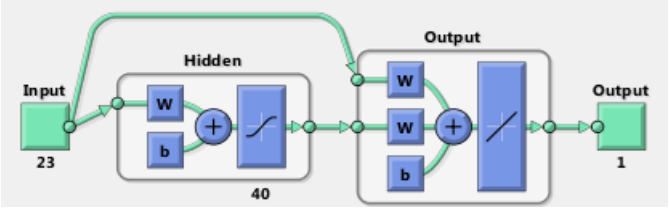

Figura 9. Red neuronal conexión entrada con capas oculta y salida

Tabla 4. Pruebas con las conexiones de las neuronas

Prueba	Conexión	Variables	Regresión
Cuarta	Entrada con capa oculta	23	0.28229
Séptima	Entrada con capa oculta y salida		0.26795
Sexta	Entrada con capa oculta	29	0.31476
Octava	Entrada con capa oculta y salida		0.32913

En la Tabla 4 se observa que no afecta igual en cada prueba y que el cambio no es muy elevado.

3.3. Función de transferencia

Se vio el efecto que tenían las 9 funciones de transferencia en la capa oculta y la de salida. En la capa de salida se dedujo que la función linear era correcta y en la capa oculta que tres funciones de transferencia podrían arrojar buenos resultados, saturating linear, hyperbolic tangent sigmoid y log-sigmoid.

La Tabla 5 refleja que estas tres funciones de transferencia reflejan unos rendimientos parecidos y a cada prueba le afecta diferente.

Tabla 5. Pruebas con funciones de transferencia

Prueba	Función de transferencia	Variables	Regresión
Primera	Tansig	16	0.38894
	Logsig		0.36896
	Satlin		0.31305
Cuarta	Tansig	23	0.28229
	Logsig		0.3117
	Satlin		0.30883
Séptima	Tansig		0.26795
	Logsig		0.27666
	Satlin		0.16433
Sexta	Tansig	29	0.31476
	Logsig		0.23619
	Satlin		0.32835
Octava	Tansig		0.32913
	Logsig		0.31444
	Satlin		0.31414

3.4. Algoritmos de entrenamiento

Al probar los distintos 12 algoritmos de entrenamiento el tiempo de entrenamiento de la red neuronal podía variar entre menos de 1 minuto hasta 10 minutos pero también variaba en gran medida el rendimiento de la red.

Los algoritmos que mejores resultados proporcionaron fueron Levenberg-Marquardt y Bayesian regularization.

La primera y octava prueba presentaban los mejores rendimientos.

En la Tabla 6 el algoritmo de entrenamiento Levenberg-Marquardt representa mayores rendimientos.

Tabla 6. Pruebas con algoritmos de entrenamiento

Prueba	Función de transf	Algoritmo	Regresión
Primera	Tansig	Trainscg	0.38894
		Trainlm	0.70651
		Trainbr	0.72334
Octava	Tansig	Trainscg	0.32913
		Trainlm	0.7488
		Trainbr	0.7073
	Logsig	Trainscg	0.31444
		Trainlm	0.79071
	Satlin	Trainscg	0.31414
		Trainlm	0.7642
		Trainbr	0.54634

3.5. Neuronas

El número de neuronas de partida no sigue un criterio único en la información consultada.

Algunos criterios de inicio:

- $n_{\text{capa oculta}} = 2 \cdot n_{\text{entrada}} + 1$
- $n_{\text{capa oculta}} = 3 \cdot n_{\text{entrada}}$
- $n_{\text{capa oculta}} = \sqrt{n_{\text{entrada}} \cdot n_{\text{salida}}}$
- $n_{\text{capa oculta}} = \frac{n_{\text{entrada}} + n_{\text{salida}}}{2}$

En un principio se tomó el primer criterio pero después se fue disminuyendo el número de neuronas en la capa oculta y se observó que el rendimiento prácticamente no se veía alterado.

En la prueba octava con el algoritmo de entrenamiento Levenberg-Marquardt se probaron distintos números de neuronas.

Tabla 7. Pruebas con el número de neuronas

Prueba	Función de transf	Número de neuronas	Regresión
Octava	Satlin	60	0.7642
		29	0.78356
	Logsig	60	0.78878
		29	0.78002

Al observar la Tabla 7 se puede deducir que el número de neuronas que se está estudiando no un gran efecto en la red neuronal.

3.6. Red neuronal

Después de analizar los resultados y datos obtenidos en las distintas pruebas realizadas se considera que la mejor red neuronal que se puede diseñar,

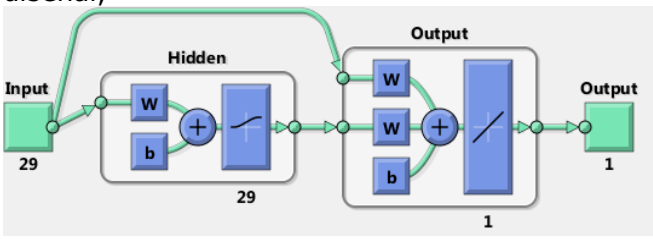

Figura 10, presenta las siguientes características:

- Entrada: 1 vector de entrada con 29 variables.
- 3 capas: 1 capa de entrada, 1 capa oculta y 1 capa de salida. La capa entrada unida a la capa oculta y la capa de salida.
- Número de neuronas: Se tomaron 29 neuronas en la capa oculta.
- Función de transferencia en la capa oculta log-sigmoid y en la capa de salida linear.
- Algoritmo de entrenamiento: Levenberg-Marquardt.

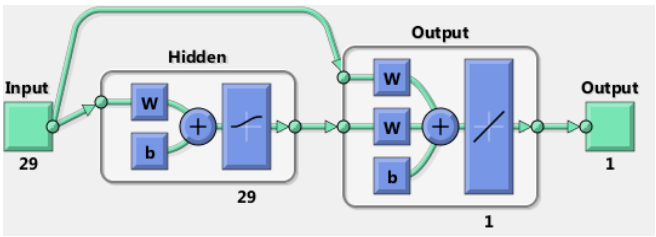

Figura 10. Red neuronal

Al observar los errores de entrenamiento, validación y testado, Figura 11, son similares lo que nos indica que no existe sobreajuste o extrapolación.

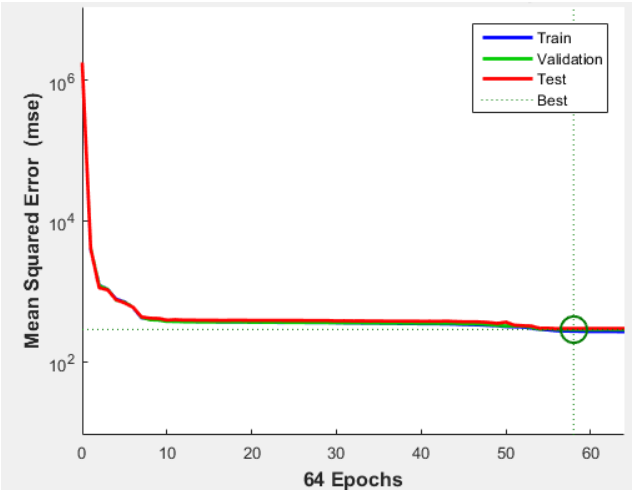

Figura 11. Errores de entrenamiento, validación y testado frente al número de iteraciones

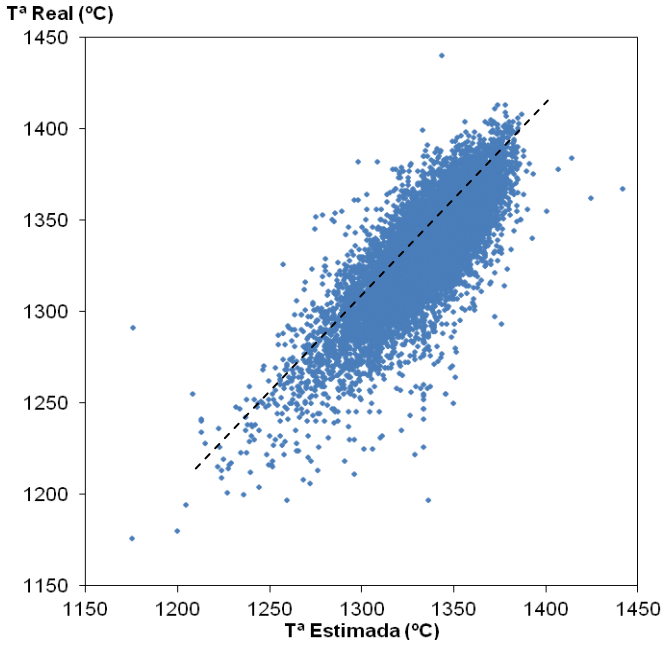

Figura 12. Resultado de comparación entre las temperaturas real y estimada

Error cuadrático medio es 16.64 ºC y la regresión 0.78. En la Figura 12 se comparan los valores de la temperatura real y la estimada.

Al observar los valores de los pesos no se encuentra ningún valor que nos haya llegar a la

conclusión de que sea necesario eliminar alguna variable de entrada. Los resultados se muestran en el anexo 4.

4. CONCLUSIONES

Las redes neuronales son una herramienta de trabajo muy útil en la búsqueda de modelos matemáticos.

Sus principales ventajas son:

- La red neuronal aprende al aportarle una serie de valores de entrada y un valor de salida esperado.
- La red se encarga de la normalización y tratamiento de las variables durante su ejecución.
- Cada red neuronal crea su propia representación de la información.
- El método presenta tolerancia a los fallos.
- Es flexible.
- Aporta datos en tiempo real.

En este trabajo se ha creado una red neuronal con una serie determinada de variables de entrada reales de los altos hornos, torpedos, cuchara y convertidor. Número de colada, de torpedo y de cuchara, temperatura del arrabio líquido procedente del alto horno y tiempos del proceso.

La temperatura a tiempo real, a día de hoy, no se puede predecir con exactitud. En este trabajo se ha buscado un modelo que permita esta predicción de manera satisfactoria.

La red neuronal nos ha aportado la información necesaria para considerar que las variables tenidas en cuenta influyen en la predicción del valor de la temperatura al convertidor.

Hay que tener en cuenta que el arrabio líquido que llega al convertidor procede de 2 o 3 torpedos distintos y de 1 o 2 hornos altos. Además los torpedos siguen una serie de pasos que influyen en su temperatura. Existe una serie de incertidumbres.

El estudio nos da una calidad que indica que se va por el buen camino. En la mejora del estudio quizás sería mejor un mayor número de variables.

En redes neuronales se pueden realizar estudios con redes dinámicas mediante análisis de series temporales que podrían complementar este estudio.

AGRADECIMIENTOS

Me gustaría agradecer a mis tutores, Don Francisco Javier Fernández García y Don José Díaz Trapiella por su trabajo y dedicación a lo largo del desarrollo de este proyecto y por ejercer de guías durante todo el proceso de elaboración que tiene como resultado el trabajo expuesto.

También mi mostrar mi agradecimiento a Arcelor Mittal.

REFERENCIAS

[1] Equipo del centro de formación Arcelor Mittal Asturias. *El proceso siderúrgico.* 2ª ed. Arcelor-Mittal, 2007

[2] Niedrighaus, J.C., Blattner, J.L. y Engel, R. Armco's experimental 184 mile hot metal shipment [en línea]. En: Ironmaking Conference Proccedings. (47º: 1988: Pennsylvania State University, USA)

[3] Verdeja González, L.F., Barbés Fernández, M.F., González Ojeda, R., Castillo, A.G. y Colás, R. Thermal modeling of a torpedo-car. *Revista de metalurgia.* 2005, 41, 449-455

[4] Chen, J. A Predective system for blast furnances by integrating a neural network with qualitive analysis. *Engineering Applications of Artificial Intelligence.* 2001, 14, 77-85

[5] Cox, I.J., Lewis, R.W., Ransing, R.S., Laszczewski, H. y Berni, G. Application of neural computing in basic oxygen steelmaking. *Journal of Materials Processing Technology.* 2002, 120, 310-315

[6] Hagan, M.T., Demuth, H.B., Beale, M.H. y De Jesús, O. *Neural network design* [en línea]. 2ª ed. Septiembre 2014

[7] Mathworks Matlab R2014a. Programa informático

[8] Castro García, J. G. *Fundamentos de la implementación de una red neuronal perceptrón multicapa mediante software.* Universidad San Carlos de Guatemala, 2006

[9] González, J.A., da Silveirlo M.A. y Pacheco, E.J. Comparación de la red neuronal y del filtro de Kalman en la estimación de velocidad del motos de inducción. *Primer congreso iberoamericano de estudiantes de ingeniería eléctrica.* 2004

[10] Martín, O., López, M. y Martín, F. redes neuronales artificiales para la predicción de la calidad en soldadura por resistencia por puntos. *Revista de metalurgia.* 2006, 42 (5), 345-353

[11] Yilmaz, F., Selbaş, R. y Şencan Şahin, A. Efficiency analysis of organic rankine cycle with internal heat exchanger using neural network. *Heat and mass transfer.* 2015

ANEXO 1. CÓDIGO MATLAB

```
    net = network
    Filtro de datos
A = xlsread('DatosTFM3','Hoja3','A2:AJ14809');
A(find((A(:,7)<0)|(A(:,8)<0)|(A(:,9)<0)|(A(:,32)<0)|(A(:,32)>60
0)),:) = [];
B = A';
    Variables de entrada
X = B(1:29,:);
    Variables de salida
T = B(30,:);
    Red 1 en entrada y 29 neuronas en capa oculta
Net = cascadeforwardnet(29);
    Número de vectores de entrada
net.numInputs=1
    Número de capas que faltan
net.numLayers=2;
net.layers{1}.size=29;
net.layers{2}.size=1;
    Conexión entre capas
net.inputConnect(1,1)=1;
net.layerConnect(2,1)=1;
net.outputConnect(1)=0;
net.outputConnect(2)=1;
    Funciones de activación (capa oculta logsig y  capa salida
purelin)
net.layers{1}.transferFcn = 'logsig';
net.layers{2}.transferFcn = 'purelin';
    Sesgo (Capas 1, 2 tienen sesgo)
net.biasConnect = [1; 1];
    Ver red
view(net)
    Inicializar pesos y sesgos de modo aleatorio
net.initFcn = 'initlay';
    Función de control mse y de entrenamiento algoritmo
Levenber-Marquardt
net.performFcn = 'mse';
net.trainFcn = 'trainlm';
net.divideFcn = 'dividerand'
    70% datos entrenamiento y 15% datos validación y prueba
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
    Función de gráficos ver estado de convergencia durante
entrenamiento
net.plotFcns = {'plotperform','plottrainstate'};
net.inputs{1}.processFcns = {'mapminmax'};
    Inicializar la red para que bias y pesos tomen un valor de
inicio
net=init(net);
    Número iteraciones, tasa aprendizaje y factor momento
net.trainParam.epochs = 100;
net.trainParam.lr = 0.05;
net.trainParam.mc = 0.95;
    Entrenar la red
[net,tr] = train(net,X,T);
    Simular la red entrenada con los valores de validación y
prueba
Y = sim(net,X);
    Salidas gráficas (muestran los valores estimados y las
salidas deseadas)
plot(Y,'o')
hold
plot(T,'*')
legend('valores                    estimados','salidas
deseadas','Location','NorthWest')
    Gráfica del funcionamiento de la red en las etapas de
entrenamiento, validación y prueba, así como el mejor resultado
plotperform(tr)
    Ajuste entre los valores estimados y los predichos
plotregression(Y,T)
    Análisis de funcionamiento con el error cuadrático medio
d = (Y-T).^2;
mse = mean(d)
    Pesos y sesgos
W = net.IW
b = net.b
```

ANEXO 2. COMPARACIÓN DE RESULTADOS

Tabla 8. Comparación de algunos valores de la temperatura medida y la estimada

	Temperatura medida (ºC)	Temperatura estimada (ºC)	Diferencia (ºC)		Temperatura medida (ºC)	Temperatura estimada (ºC)	Diferencia (ºC)
1	1290	1299	9	26	1305	1311	6
2	1319	1313	-6	27	1337	1351	14
3	1305	1312	7	28	1361	1364	3
4	1302	1319	17	29	1333	1333	0
5	1315	1343	28	30	1355	1369	14
6	1311	1307	-4	31	1341	1347	6
7	1314	1319	5	32	1372	1355	-17
8	1298	1316	18	33	1357	1359	2
9	1295	1311	6	34	1345	1343	-2
10	1325	1325	0	35	1359	1355	-4
11	1301	1322	21	36	1364	1353	-11
12	1328	1344	16	37	1355	1352	-3
13	1312	1340	28	38	1342	1350	8
14	1310	1327	17	39	1381	1369	-12
15	1355	1354	-1	40	1336	1344	8
16	1334	1336	2	41	1348	1349	1
17	1328	1341	13	42	1343	1351	8
18	1334	1350	16	43	1321	1353	32
19	1327	1328	1	44	1348	1338	-10
20	1350	1338	-12	45	1328	1342	14
21	1347	1362	15	46	1351	1342	-9
22	1345	1333	-12	47	1335	1348	13
23	1331	1349	18	48	1352	1351	-1
24	1369	1337	-32	49	1341	1335	-6
25	1350	1340	-10	50	1349	1349	0

ANEXO 3 TABLAS DE PESOS Y SESGOS

Tabla 9. Matriz de pesos de la capa oculta

	1	2	3	4	5	6	7	8	9	10
1	-1.826	10.825	31.795	2.075	-1.375	-2.558	-0.574	-14.158	-16.257	-10.853
2	-17.462	-23.108	16.431	28.599	0.735	-9.311	21.014	20.604	25.151	29.699
3	4.631	1.112	-4.824	3.967	-0.099	-3.861	14.839	5.788	-1.915	-2.982
4	-31.732	-41.014	8.837	-29.361	-29.759	-4.050	-0.371	-14.590	20.545	-32.597
5	-27.491	-12.298	-3.940	-13.707	103.57	-20.386	24.327	-21.418	-21.059	-42.194
6	2.714	1.827	-1.865	-4.783	0.070	-1.967	-0.166	2.680	-7.370	-4.020
7	-0.709	-5.871	13.246	-1.616	-1.263	-1.854	-0.522	-2.508	-9.150	-2.950
8	-26.988	-28.390	59.772	-39.999	-11.238	11.080	41.132	-43.543	100.38	35.575
9	-0.249	-1.063	12.919	-0.205	-0.377	-1.579	0.010	8.294	7.857	8.344
10	0.086	-0.066	0.500	-0.019	-0.007	-0.370	-0.076	-0.751	0.598	0.519
11	10.494	17.108	-38.329	-18.791	-1.789	6.218	12.355	-38.659	-38.350	-0.741
12	-1.066	0.204	-2.708	0.488	1.657	0.369	-2.613	4.144	0.256	0.050
13	-29.674	-33.271	66.121	9.106	4.528	45.331	-16.836	28.962	36.030	51.804
14	-1.155	1.829	1.399	-2.678	1.437	0.299	8.260	-2.146	-11.926	-2.023
15	2.204	5.642	2.811	-2.261	5.885	-2.810	6.507	-5.434	4.449	-6.109
16	4.680	-27.424	-53.830	-7.678	8.688	20.162	3.629	-30.450	-0.840	-9.157
17	-2.391	-4.545	3.180	-3.384	-2.374	0.700	2.584	-2.016	-0.657	-1.031
18	-2.727	-3.735	-2.897	-1.361	-0.832	0.640	-0.029	4.634	-3.275	3.517
19	0.034	0.318	-0.735	0.017	-0.220	-0.406	-0.280	-0.811	-0.052	0.740
20	-3.559	0.579	-0.019	1.189	0.912	11.304	13.622	3.235	0.019	-24.907
21	-4.315	-3.972	5.700	-1.732	0.291	4.800	-4.354	-0.589	-1.373	2.331
22	2.718	-0.190	-3.174	-1.480	-3.453	-8.456	-15.907	-5.460	0.987	-3.727
23	-0.210	1.564	-2.233	-0.152	0.006	-2.031	-0.020	-1.841	-1.961	-1.059
24	2.797	-1.332	-1.833	-0.557	-5.215	0.027	0.219	-2.461	6.947	1.289
25	-1.487	-0.171	0.205	0.819	0.561	2.338	-0.015	-0.753	0.644	2.594
26	2.165	1.964	-3.271	1.866	-2.103	-5.032	3.460	1.537	5.047	-4.359
27	0.147	0.051	0.543	0.013	-0.053	-0.125	-0.032	0.505	0.429	-0.960
28	-10.395	6.054	5.167	-2.151	3.634	-6.883	2.550	-12.980	-6.839	-11.097
29	-0.016	-0.027	-2.551	-0.056	-0.363	0.097	-0.326	2.808	0.814	2.505

	11	12	13	14	15	16	17	18	19	20
1	-61.435	37.194	164.18	36.493	-10.964	0.539	35.414	43.990	-9.797	35.782
2	34.246	38.793	43.341	47.888	52.436	56.983	61.530	66.078	70.625	75.173
3	-0.450	-3.544	-3.319	-2.466	-1.943	-5.092	-5.798	-1.019	-7.346	-7.132
4	-25.361	11.384	22.574	29.372	28.192	-24.590	-16.434	-43.638	31.260	-34.565
5	-45.411	-43.379	16.942	1.294	-11.282	27.521	-5.924	-16.872	-13.413	-15.369
6	2.298	1.445	-2.336	-5.432	-3.396	-3.406	-5.170	-2.055	-1.008	-4.877
7	-2.171	-47.265	-2.710	-21.557	21.476	-1.504	-4.648	-0.097	-10.663	1.580
8	18.126	18.546	20.947	-0.978	4.573	-4.284	41.797	6.404	9.612	42.206
9	0.412	-8.111	-25.397	13.185	14.566	5.460	-99.458	-37.468	60.304	-74.040
10	-1.008	14.136	-0.608	-0.510	-3.414	-0.132	52.075	-65.655	64.322	-51.620
11	23.011	-30.614	11.544	-11.709	16.050	-7.228	-24.273	31.092	18.991	-35.715
12	0.199	-0.402	-1.890	2.437	2.124	2.852	-0.296	3.178	1.913	-0.308
13	144.05	19.447	65.917	-121.64	-32.983	45.554	31.720	-84.413	34.567	27.845
14	6.996	4.230	2.020	-6.206	-1.490	-5.704	-9.110	-2.484	-0.877	-9.037
15	-0.767	3.243	3.861	-5.992	-1.110	-2.694	-3.377	-5.916	1.178	-3.347
16	-46.970	-44.183	32.407	-19.100	11.517	81.269	-45.489	8.118	-11.61	-46.004
17	-2.277	-2.378	2.572	-4.607	-1.230	-2.283	-3.002	-1.758	-3.765	-3.123
18	-1.431	-1.458	-3.032	2.831	3.082	1.613	1.009	3.285	1.612	0.023
19	23.108	10.076	1.522	-1.275	-1.738	-0.096	35.782	2.477	4.040	-80.095
20	-26.148	-3.456	20.056	-30.116	19.026	-32.169	42.761	-72.832	90.782	39.919
21	2.885	1.544	4.127	2.897	4.208	4.447	3.633	3.689	5.129	3.735
22	-6.046	-2.130	-4.777	-8.482	-8.663	-10.052	-3.880	-0.290	-8.198	-4.031
23	0.950	0.537	2.232	-0.644	-0.773	0.659	-3.957	20.344	14.398	6.511
24	-0.262	-0.757	-1.631	0.637	-0.490	0.486	1.751	4.503	0.305	2.697
25	-0.535	-0.539	0.905	4.268	2.462	2.623	-2.896	5.590	2.372	-1.370
26	-4.341	-2.582	-3.575	-2.001	-4.025	-3.090	-5.193	-2.341	-5.991	-3.304
27	-1.543	0.228	-0.796	1.183	1.321	0.189	61.951	63.958	46.505	-61.806
28	38.738	-68.972	-5.573	-62.774	13.500	7.654	-26.692	-40.348	7.173	-42.030
29	29.783	-9.689	3.419	-7.074	3.342	0.023	-75.799	36.623	3.675	-95.973

	21	22	23	24	25	26	27	28	29
1	47.689	-15.156	-11.822	25.181	19.391	2.975	-12.904	-23.109	-20.605
2	-4.654	-4.857	15.207	-16.311	12.553	-19.046	9.910	5.390	-20.362
3	-1.505	-6.997	-0.236	-0.218	-6.300	-1.826	-1.141	-7.632	-0.351
4	-64.216	31.556	22.637	-3.447	-9.965	-62.638	64.271	-65.514	-9.729
5	-28.081	-13.791	44.929	15.289	31.123	-51.747	-37.450	-14.743	-2.251
6	-1.972	-1.140	-6.877	-4.557	-5.133	-5.856	-3.212	-3.870	-5.321
7	4.772	-0.993	-65.819	-4.964	25.320	24.155	-8.871	57.170	3.363
8	7.447	10.086	6.305	-13.253	-41.921	2.699	2.491	25.884	7.059
9	-29.390	61.966	-41.697	-35.046	8.890	31.129	37.594	27.866	-3.872
10	13.732	-63.876	4.812	-5.544	0.275	17.852	5.592	0.169	3.259
11	19.475	16.412	-30.605	-9.665	14.117	87.366	-40.010	-24.185	-1.134
12	2.688	1.479	3.668	3.863	2.482	2.567	2.215	3.325	2.814
13	-88.133	33.371	-161.88	-22.064	36.250	-88.616	11.062	36.258	31.659
14	-1.919	-1.595	-8.598	-2.686	-6.741	-8.137	-1.776	-6.123	-6.077
15	-7.153	0.811	-5.797	-2.450	-5.069	-6.998	-1.102	-5.691	-5.480
16	8.522	-12.573	-7.615	18.830	-23.048	0.908	9.932	-4.258	-9.699
17	-1.404	-3.371	-3.151	-0.218	-4.060	-5.656	0.450	-2.730	-4.917
18	4.048	2.567	4.570	3.057	1.915	2.982	2.602	2.790	2.783
19	-16.670	-54.937	17.197	16.513	-16.83	-31.077	-21.309	-2.052	-0.220
20	-71.275	89.540	46.874	-3.234	-113.31	-14.702	-12.916	-45.022	-10.322
21	4.866	5.399	1.736	4.513	5.552	2.449	3.871	4.047	2.791
22	0.785	-9.324	-7.881	-7.264	-10.97	-7.061	-9.409	-11.050	-8.416
23	-17.066	-8.609	-5.623	2.743	-7.265	13.694	5.469	-53.065	-123.04
24	5.176	0.344	-0.013	0.150	0.136	1.489	0.589	0.519	0.539
25	4.678	2.601	4.308	3.054	3.157	2.904	3.782	2.359	3.857
26	-3.683	-5.752	-2.185	-4.262	-5.707	-1.178	-4.586	-5.249	-1.978
27	-66.265	-26.697	-31.729	44.668	27.310	-656.71	-451.31	-4.384	4.503
28	-60.990	-3.913	86.585	33.067	40.680	-64.455	42.926	-10.658	8.571
29	8.418	-1.102	33.266	9.301	11.706	-63.613	21.456	-34.941	1.830

Tabla 10. Pesos de la capa de salida

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1,286	-17,092	-15,862	1,303	4,525	18,312	4,605	-2,685	-37,760	-32,263
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
4,282	-30,267	43,117	54,485	45,873	-2,015	1,179	15,827	-45,297	-6,026
21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	
7,963	-53,631	-165,72	-47,873	-66,649	-201,97	9,789	46,345	-95,953	

Tabla 11. Sesgos de la capa oculta

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
-2.070	8.506	4.047	-33.258	-16.407	3.806	28.717	-10.976	-57.125	-31.739
11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
-42.971	-2.590	-42.432	7.085	6.467	3.688	3.954	-3.196	-154.23	-69.943
21	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	
-3.973	9.498	-162.93	-2.858	-5.589	-0.533	-1048.6	-69.329	-149.34	

Tabla 12. Sesgo de la capa de salida

1
133.589