

---

# IDENTIFICACIÓN DE PATRONES EN SERIES DE TIEMPO USANDO REDES NEURONALES EN DATOS DE UNA EMPRESA PETROQUÍMICA

---

LEONARDO BUSCHIAZZO\*  
LORENA PRADENAS\*\*

## Resumen

*En este artículo se presenta un estudio para la identificación de patrones en series de tiempo de datos de proceso basado en redes neuronales, con el objetivo de apoyar a la gestión del control automático en una industria de procesos continua. Para implementar la identificación de patrones, se realizó un sistema computacional que contiene un algoritmo de redes neuronales para la clasificación de segmentos de una serie de tiempo en casos conocidos predefinidos. Así, con el sistema computacional se evaluó la temperatura de salida del producto de un horno automatizado de una empresa petroquímica de la octava región, considerando un mes de datos en crudo y filtrados. Al clasificar los patrones de la salida del horno, se obtuvo un rendimiento general de un 83,33% para los datos crudos y un 82,96% para los datos filtrados, para patrones conocidos y desconocidos. En el caso particular de los datos conocidos, la aplicación tuvo un rendimiento de clasificación similar para entradas filtradas y no filtradas, lo que es un buen indicador acerca de la tolerancia de la red a conjuntos de datos sin filtrar. A partir de los resultados y de la experiencia del estudio, se concluyó que es válido el uso de la herramienta para clasificación de patrones, siendo un apoyo para la evaluación del comportamiento del control automático de un proceso, además, esta herramienta es aplicable como apoyo en algunos tipos de cartas de control ocupadas en gestión de calidad. En cuanto a trabajos futuros, se puede mencionar que la herramienta es perfectible en cuanto a un mayor tipo de patrones de entrenamiento, arquitecturas de redes neuronales, mejores datos de entrenamiento y la implementación de un esquema que permita evaluar en tiempo real el proceso en estudio para propósito de alarmas de proceso.*

**Palabras Clave:** Redes Neuronales Artificiales, Patrones, Procesos Continuos, Series de tiempo.

---

\*Magíster en Ingeniería Industrial, Universidad de Concepción.

\*\*Departamento Ingeniería Industrial, Universidad de Concepción.

---

## 1. Introducción

---

En las industrias de procesos continuos y semi continuos es común el uso de tecnologías de automatización para el control de la producción. Existen empresas donde se han implementado sistemas de control distribuido (conocidos como DCS por sus siglas en inglés) o sistemas de Control supervisor y adquisición de datos (conocidos como SCADA, por sus siglas en inglés) para manejar los principales procesos productivos. Esto implica mejoras en la calidad del proceso, ahorros en costos de producción, manejo adecuado de los tiempos de producción y, por ende, de logística, entre otras ventajas con respecto a una producción realizada en forma manual por personal capacitado. En otras palabras, implementar automatización aumenta cuantitativamente la competitividad de una empresa. Sin embargo esto es en el escenario ideal donde el control automático funciona sin problemas y maneja el proceso de producción tal que su salida es homogénea y acotada alrededor de un valor objetivo deseado. En el escenario real, el control automático del proceso no es infalible y es propenso a perturbaciones, debido a causas al azar o causas especiales (Gutiérrez [4]), que provocan un comportamiento anormal en la salida del proceso controlado.

Esta situación lleva a plantearse varias preguntas: ¿Quién controla al controlador, en cuanto a que realice su labor apropiadamente?, ¿Cómo detectar cambios en el proceso que no sean gruesos pero que implican una anomalía? y ¿Cómo realizar la detección en forma oportuna? o visto desde otro punto de vista, cómo realizar una gestión operativa de estos sistemas de forma cuantitativa y objetiva.

Entonces, a modo de respuesta de las interrogantes planteadas, se propuso que un camino para apoyar a la gestión operativa de un sistema de control automático es por medio de la revisión el patrón de salida en el tiempo de la variable del proceso que se controla. Un comportamiento normal de la salida del proceso tiene un patrón de salida característico, luego, otros patrones de salida implican la existencia de alguna anomalía en la capacidad del sistema de control para manejar el proceso.

Para realizar la detección, se planteó el desarrollo de una herramienta computacional que tuviera como núcleo un modelo de redes de neuronas artificiales. Modelo que tienen una amplia experiencia en la clasificación de elementos de un mismo conjunto (Matich [10], Viñuela [22], Hann [6], por mencionar algunos autores) sin la necesidad de entender como se originan éstos, sino que aprende por medio de ejemplo de elementos ya existentes y que se cuentan con variadas experiencias en el trabajo con Patrones de datos (Fu [3], Guh [5], Olmedo [15] y Singh [18], entre otros) y en la aplicación en el ámbito de

sistemas industriales (Muñoz [13], Silva [19], Ruiz [16], entre muchos otros).

Se trabajó con datos de un proceso productivo real de una empresa de la octava región. Al ser datos de terreno, éstos contienen datos poco confiables por que se definió evaluar la herramienta computacional con los datos del proceso en forma directa y con los mismos datos pero con un pre procesamiento para descartar valores poco confiables (picos en la señal principalmente).

En la siguiente sección se presenta una introducción al problema en estudio, describiendo un proceso real donde se evaluó la solución propuesta. Luego, se describe el diseño de la solución propuesta, se detalla el sistema computacional desarrollado específicamente para implementar el modelo de solución. A continuación se muestran los principales resultados obtenidos al evaluar datos del proceso real de estudio con la herramienta computacional. Finalmente, se dan a conocer las conclusiones obtenidas en base a los resultados y se presentan una serie de propuestas para trabajos futuros.

---

## 2. Problema en Estudio

---

El problema de investigación es la detección de patrones en una serie de tiempo para una variable controlada de un proceso real de producción industrial por medio de un sistema computacional. En particular, la variable controlada es la salida de un proceso productivo en una planta continua. Básicamente, se tiene un patrón de señal “normal” y patrones que se alejan de una distribución normal, lo que indica la presencia de alguna perturbación que no ha podido ser compensada por el ente controlador.

Así, un patrón normal de la variable controlada en el tiempo sería como el ilustrado en la figura 1, donde la señal varía alrededor de un valor objetivo.

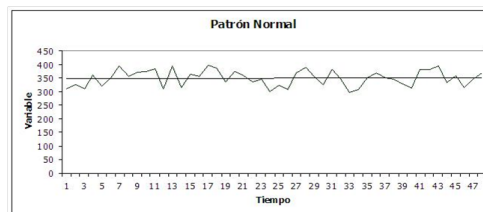


Figura 1: Patrón normal.

Por otro lado, comportamientos anormales en la salida serían los ilustrados en las figuras 2 y 3. En la figura 2 se tiene un salto (escalón) de un valor objetivo a otro y en la figura 3 un patrón creciente, que se va alejando del valor objetivo a través de una recta de pendiente positiva.

Los escalones pueden ser originados debido a un cambio operacional del proceso lo cual es una causa normal, por otro lado, si no se tenía programado,



Figura 2: Patrón escalón positivo.

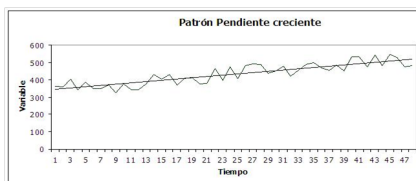


Figura 3: Patrón de pendiente creciente.

es una anomalía de la operación del control.

### 2.1. Proceso en Estudio

El estudio se realizó en un proceso real, de una empresa de la región del Bío-Bío, correspondiendo a un horno de calentamiento de materia prima, donde se controla la temperatura de salida de la materia de forma automática por medio de un sistema mecánico-computacional dedicado (DCS). La figura 4 muestra un esquema del proceso, donde se aprecia la entrada y salida de materia prima, líneas verdes continuas desde el tren de crudo hasta el equipo E-1; varias líneas de alimentación del horno (aire, fuel gas, fuel oil) y las líneas punteadas que representan la red de control sobre la temperatura de salida del horno.

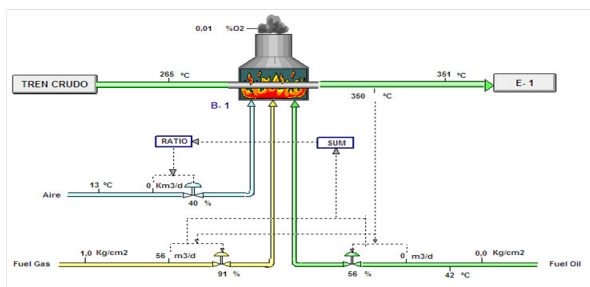


Figura 4: Horno de precalentamiento de materia prima.

Se pretende mantener la temperatura de salida de la materia prima en un valor cercano al valor definido por el departamento de operaciones de la empresa como el óptimo. El sistema de control manipula una serie de entradas a los quemadores del horno, tal que se aumente, disminuya o mantenga la temperatura de salida.

Al ser un proceso real, en la adquisición de datos es inherente tener datos poco confiables, debido al ruido eléctrico que afecta a los instrumentos de terreno. Dado esto, se trabajó con dos grupos de datos. Un grupo corresponde a los datos obtenidos de terreno y otro corresponde a los mismos datos de terreno pero con un pre-procesamiento para prevenir datos no confiables por medio de un filtro de suavizamiento exponencial.

### 3. Solución

#### 3.1. Red de neuronas artificial

El problema a resolver es diseñar un sistema que tuviera como entradas el muestreo discreto de una señal de proceso en el tiempo, realizado en ventanas fijas de tiempo, como se muestra en la figura 5, y como salida una clasificación de la forma de la señal ingresada. Se diseñó un sistema computacional con un algoritmo de redes neuronales como núcleo, con la capacidad de reconocer patrones en las señales discretas ingresadas.

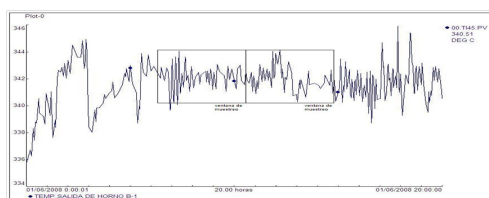


Figura 5: Serie de tiempo de variable de proceso.

El modelo de red neuronal seleccionado es una arquitectura de tipo Perceptron Multi Capa, el cual es un modelo ampliamente difundido y probado en lo relativo a la clasificación de entradas en un red ([22], [17], [5], [6], entre otros).

En la figura 6 se ilustra un diagrama de entradas y salidas, con el diseño del proceso de clasificación de patrones con la red neuronal como núcleo de procesamiento. En esta el muestreo para un periodo de tiempo fijo (ventana de tiempo) de la tendencia en el tiempo de la variable de proceso es ingresado como un vector de valores a la red; siendo procesado por ésta, entregando como salida de la operación un vector de elementos binarios que corresponde al código del patrón que mejor se ajusta a la forma de la señal de entrada.

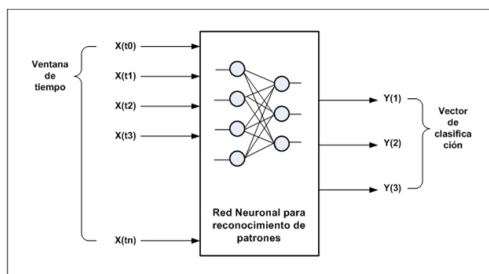


Figura 6: Red Neuronal Artificial como solución para el problema.

La red tiene como entrada un vector que representa el muestreo discreto en el tiempo de una señal de una variable de proceso controlada. El número de elementos del vector de entrada es fijo y se inicia desde el primer valor en la ventana de tiempo analizada hasta el último valor de la ventana de tiempo, es decir, el vector es la segmentación de la serie de tiempo dentro de un marco de tiempo fijo. Los valores en sí, corresponden a variables numéricas de tipo real. Dado esto, y sumado el hecho de que la red sólo puede tener entradas reales entre valores comprendidos en el siguiente intervalo  $[-1, 1]$ , se consideró una transformación lineal de los valores del vector de entrada de modo de acotar el rango permitido para la red. Esta transformación se implementa en el sistema computacional.

La salida de la red es un vector que indica a que patrón se ajusta mejor la entrada, según los patrones que la red ha aprendido previamente en su etapa de entrenamiento. La salida está compuesta por un vector de tres elementos, donde los elementos son variables de tipo binario. Se tiene un conjunto de combinaciones del grupo de elementos para definir un tipo de patrón en particular. Así, se definieron los patrones que aparecen en la tabla 1. Con estos patrones se entrenó la red de modo que fuera capaz de reconocer cuando alguno de ellos se encuentra presente en una tendencia de una variable.

Nombre del Patrón	Clase		
	Y(1)	Y(2)	Y(3)
Normal	0	0	1
Pendiente Creciente	0	1	0
Pendiente Decreciente	0	1	1
Escalón positivo	1	0	0
Escalón negativo	1	1	0

Tabla 1: Patrones definidos para el estudio.

### 3.1.1. Datos de entrenamiento y validación

Los datos para entrenar la red y para su validación, se generaron a partir de un conjunto de fórmulas propuestas en [17] y que ocupan información estadística de valor verdadero y de la desviación estándar de la variable en estudio. Las fórmulas utilizadas son las (1) a (5).

- (1) Patrón Normal  $f(x) = \eta + r(x) \cdot \sigma \quad \dots (1)$
- (2) Patrón de Pendiente Creciente  $f(x) = \eta + r(x) \cdot \sigma + g \cdot x \quad \dots (2)$
- (3) Patrón de Pendiente Decreciente  $f(x) = \eta + r(x) \cdot \sigma - g \cdot x \quad \dots (3)$
- (4) Patrón de Escalón positivo  $f(x) = \eta + r(x) \cdot \sigma + b \cdot s \quad \dots (4)$
- (5) Patrón de Escalón negativo  $f(x) = \eta + r(x) \cdot \sigma - b \cdot s \quad \dots (5)$

Donde:

- $\eta$ : es el valor verdadero de la variable de proceso para un año de datos.
- $\sigma$ : es la desviación estándar de la variable de proceso para un año de datos.
- $g$ : es el gradiente de la pendiente en ecuación (2) y (3).
- $b$ : es el pivote para el escalón, sus valores posibles son 0 y 1. En el caso de la ecuación (4), toma el valor 0 hasta el valor  $x$  donde se desea el escalón, luego toma el valor 1, el caso inverso se da para la ecuación (5).
- $s$ : es el valor de la magnitud del escalón de las ecuaciones (4) y (5).

Con las ecuaciones (1) a (5) se generó un conjunto de patrones de entrenamiento y validación para la red. En total se crearon 200 vectores de 48 elementos para cada tipo de patrón, 150 para entrenamiento y 50 para validación del entrenamiento. Las figuras 7, 8, 9, 10 y 11 muestran gráficos de ejemplo para cada tipo de patrón mencionado.

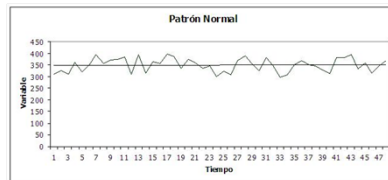


Figura 7: Patrón normal.

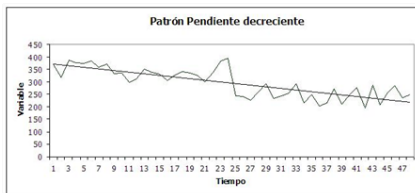


Figura 8: Patrón de pendiente decreciente.

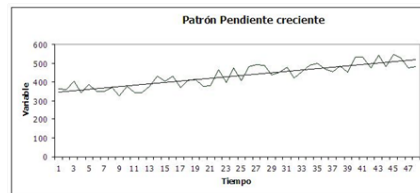


Figura 9: Patrón de pendiente creciente.

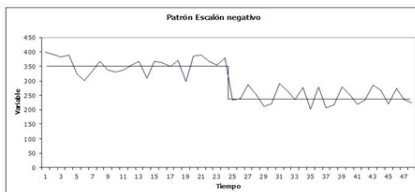


Figura 10: Patrón de escalón negativo.

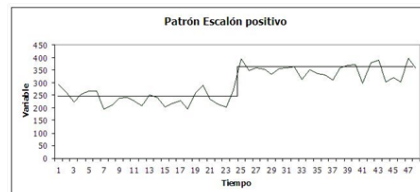


Figura 11: Patrón de escalón positivo.

### 3.1.2. Filtrado de Datos

Debido a que los datos del proceso real son obtenidos desde sensores de terreno, están expuestos a distorsiones que implican valores poco confiables en éstos, [21] y [14]. Así, para filtrar los datos, se utilizó un algoritmo de suavizamiento exponencial de primer orden, según antecedentes en [21]. Así, se suavizan los picos de datos poco confiables y la señal en general, para efectos del ruido inherente. A continuación se entrega la ecuación que define el suavizamiento exponencial, [23].

$$A_t = \alpha \cdot x_t + (1 - \alpha)A_{t-1} \quad \dots(6)$$

Donde:

- $A_t$ : valor pronosticado para la señal en el instante  $t$ . Variable dependiente.
- $A_{t-1}$ : valor pronosticado para la señal en el instante anterior a  $t$ ,  $(t - 1)$ .
- $\alpha$ : constante de suavizamiento, con valores dentro del rango  $(0, 1)$ .
- $x_t$ : valor muestreado de la señal en el instante  $t$ .

### 3.1.3. Pseudo algoritmo de entrenamiento de la Red Neuronal

El entrenamiento es el proceso por el cual se fijan los pesos de la red, es un proceso iterativo en busca de los mejores valores para estos pesos tal que se obtenga el menor error de entrenamiento y de validación de la red.

El pseudo algoritmo usado para el entrenamiento de la red se muestra a continuación:

```

Inicio;
Se inicializan los pesos y umbrales en la red en forma aleatoria;
Repetir{iniciar contador de repeticiones en n=1
  Repetir{ iniciar contador de repeticiones en i=1
    Calcular  $Y(i) = ANN(X(i));$ 
    Calcular  $E(i) = \text{Diferencia entre } S(i) \text{ y } Y(i);$ 
    Aplicar regla delta para modificar los pesos y umbrales;
  }hasta  $i=n;$ 
  Evaluar  $E_t;$ 
  Repetir{ iniciar contador de repeticiones en j=1
    Calcular  $Z(i) = ANN(W(i));$ 
    Calcular  $Ev(i)$ {
      Si  $Z(i) = V(i)$  entonces  $Ev(i) = 0$ 
      Sino  $Ev(i) = 1$ 
    }
  }hasta  $j=m;$ 
  Evaluar  $E_{vt}$  como porcentaje de aciertos del total (m);
} Hasta  $E_t < \text{Cota de Error o } n = \text{máximo de repeticiones permitido;}$ 
Fin;

```

Donde:

- El par  $(X(i), S(i))$  corresponde a un vector de parámetros de entrenamiento  $(X)$  con su respectiva salida  $(S)$ .



- $ANN()$  es la función de transferencia de la red neuronal, es decir, la propagación del vector  $X$  por la red, dando como resultado el vector  $Y$ .
- $E$  es el error para la evaluación de un patrón de entrenamiento en particular y es el error cuadrático medio entre la salida conocida del patrón y la propuesta por la red.
- $E_t$  es el error total de un ciclo completo de entrenamiento.
- $W(i)$  es el  $i$ -ésimo vector de entrada con un patrón de validación.
- $Z(i)$  es el  $i$ -ésimo resultado entregado por la red para un patrón de validación.
- $E_v$  es un vector traspuesto que almacena dos valores, 0 ó 1. donde cero implica que la validación no fue exitosa y 1 que si lo fue. Esto es debido a que se trata de clasificar el patrón dentro de un conjunto de formas conocidas por la red, ergo, las opciones son 100% o 0%.
- $E_{vt}$  es el porcentaje de validaciones acertadas con respecto al número total de elementos del conjunto de validación.

**3.1.4. Diagrama resumen de la solución implementada**

A modo de consolidar la solución, en la figura 12 se muestra un diagrama que resume la solución implementada. Así, la serie de tiempo es segmentada en forma secuencial, luego, un segmento de la serie de tiempo es ingresado como un vector discreto a un algoritmo de clasificación en patrones, el cual es la red de neuronas artificial, para ser clasificado en alguna de las clases conocidas de formas de onda de serie de tiempo. La clasificación es entregada como una codificación en un vector binario.

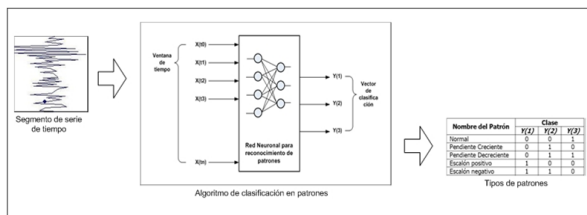


Figura 12: Diagrama conceptual de la solución.

**3.2. Sistema computacional**

Como ya se mencionó, para poder llevar a cabo el estudio en la práctica, se debió construir un sistema computacional que permitiera recopilar la información de procesos, formatearla y evaluarla según el algoritmo de clasificación basado en redes neuronales. Así, se diseñó un sistema computacional

conformado por tres entidades que interactúan entre ellas, donde se tiene un repositorio de datos de proceso en tiempo real e históricos, denominado PI System; planillas Excel con lógica imbuida para la recolección de los datos desde el repositorio y para formatear los vectores de entrada que son clasificados por un programa, de desarrollo especial para el estudio, que contiene el algoritmo de la red neuronal para clasificar la entrada en alguno de los patrones previamente definidos.

En cuanto a PI system, este es un sistema de almacenamiento de datos en tiempo real, orientado principalmente para manejar una gran cantidad de datos de variables de procesos productivos (hasta 1 millón de variables) y consiste en un servidor con una base de datos en tiempo real e histórico que se conecta a la mayoría de los sistemas de control automático del mercado, de esta forma concentra la información de una o varias plantas productivas en un único repositorio de datos. Para este trabajo se ocupó el repositorio para consultar información por la variable de temperatura de salida de un horno bajo control, su identificación es “00:ti45.pv” que es un tag de control.

Excel se utilizó para la gestión de los datos a ser ingresados al programa clasificador y para la visualización del resultado de la clasificación y como graficador. De tal modo se desarrollaron dos planillas con lógica imbuida (código Visual Basic para Aplicaciones) en este programa; una para extraer los datos de proceso desde el sistema PI y preparar archivos de entrada para el programa que los procesa y clasifica en algún patrón conocido con un algoritmo de redes neuronales y la otra planilla se usó para crear los patrones de entrenamiento y validación por medio del uso de las formulas descritas en el punto 3.1.1. El uso de Excel fue de carácter mandatorio, puesto que para poder acceder a los datos era necesario tener instalado en este programa un complemento dedicado a la conectividad con PI. Además, la facilidad que entrega Excel para estructurar los datos en matrices y graficar series de éstos es un valor agregado a tomar en cuenta.

El programa que clasifica los patrones de entradas en alguno del conjunto de los conocidos por él es una aplicación de consola sin interfaz gráfica para el usuario que lee archivos de datos de entrada tanto como para la evaluación de patrones como para el entrenamiento de la red. Los resultados los entrega como archivos compatibles con Excel, de modo de usar este último programa para revisar los resultados y graficar. Este programa corresponde a un desarrollo específico para este estudio, siendo construido bajo el lenguaje de programación Visual C# y denominado “AnnCore”. El lenguaje de programación fue seleccionado luego de revisar la literatura relacionada, destacándose los autores Chesnokov [1], Madhusudanan [8] y Kirillov [7], y encontrando que este lenguaje está totalmente vigente y conteniendo los desarrollos más recientes. Dado que es un lenguaje relativamente nuevo, se debió aprender su uso, principalmente por medio de documentación del proveedor, [11] y [12].

### 3.2.1. Arquitectura

La arquitectura de red (mostrada en la figura 13), donde se apoya el sistema computacional, es de dos capas y de tipo cliente-servidor en cuanto al flujo de datos. Así se tiene una red de control del proceso productivo, que corresponde a la primera capa, donde se origina la información y es enviada al repositorio de datos históricos y en línea de producción (PI System) por medio de una interfaz entre esta red de control y la red computacional donde se encuentran los demás equipos. En esta red computacional, segunda capa, se encuentra el servidor PI y el equipo donde reside la aplicación Excel y el programa clasificador de redes neuronales.

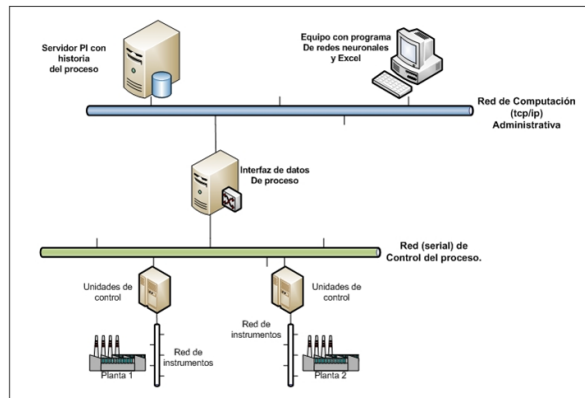


Figura 13: Arquitectura física del sistema computacional.

Para efectos de la aplicación de redes neuronales, tienen injerencia directa el servidor PI y el equipo tipo PC que contiene Excel y el programa de redes neuronales. Así, Excel saca datos desde el servidor PI por medio de librerías de comunicación, sobre TCP/IP, propietarias de PI. Luego, el procesamiento de los datos se realiza en forma local en el equipo PC.

### 3.2.2. Diagrama de Procesos Principales

La figura 14 ilustra los procesos principales en el sistema y su interacción entre estos y con los resultados. Se tiene el repositorio de datos de producción PI desde donde Excel rescata esta información y la prepara para ser ingresada al programa "AnnCore" para ser procesada.

Excel colecta la información desde el repositorio PI por medio de un complemento especial, luego, prepara la información para ser ingresada en el programa de clasificación. Una vez que este programa realiza la identificación de patrones, el resultado se visualiza en Excel.

El programa denominado "AnnCore" es un desarrollo específico para este estudio y es el que contiene el algoritmo matemático de la red de neuronas para la clasificación en patrones la serie de tiempo ingresada en forma discreta.

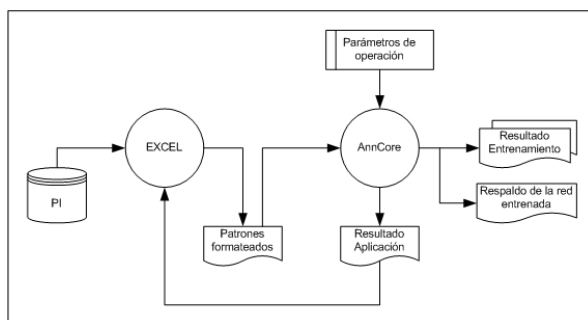


Figura 14: Diagrama de procesos principales.

Tiene dos modos de funcionamiento; uno de entrenamiento para sintonizar la red con la mejor combinación de parámetros con respecto al error de validación y entrenamiento y otro de ejecución que es para la clasificación de patrones de entrada.

En el modo de entrenamiento se ingresan patrones conocidos con sus respectivas salidas, también conocidas, para que el programa corra un algoritmo de entrenamiento de modo que la red aprenda a identificar los diferentes patrones por medio de ejemplos conocidos. Al finalizar el proceso, entrega como resultados archivos con el error de entrenamiento y validación y un archivo binario con el respaldo de la red ya entrenada, para ser usada en el modo de ejecución posteriormente.

En el modo de ejecución, el programa evalúa una serie de tiempo ingresada como un vector discreto y la clasifica según los patrones ya conocidos por éste o, en caso que no coincida con ninguno, en desconocido. El resultado es entregado como un archivo con formato compatible con Excel para su visualización.

---

## 4. Resultados

---

### 4.1. Resultados del Entrenamiento de la Red

Para que la red neuronal adquiriera la inteligencia necesaria para clasificar patrones en series de tiempo, se debió “educar” por medio de un entrenamiento repetitivo. Entonces, se realizaron una serie de pruebas, alterando los parámetros constitutivos y de entrenamiento de la red, buscando la combinación de pesos en la red que entregara el mayor porcentaje de aciertos en la validación de la red y el menor error de entrenamiento, siendo el primer punto el preponderante a la hora de la selección de la mejor configuración. El porcentaje de acierto de validación corresponde a la razón entre el número de patrones identificados correctamente y el número total de patrones a validar, en términos porcentuales, donde lo que se deseaba era obtener un porcentaje cercano

al 100 %. En el caso del error de entrenamiento, expresado como cuadrático medio, éste corresponde al error generado entre la salida dada por la red para un vector de entrada y el vector de salida conocido para aquella entrada.

En cuanto al número de pruebas, 46 en total, se contrastó con experiencias de autores (Sagioglu [17]; Guh [5], Hann [6]; entre otros) en diversos documentos técnicos y libros encontrándose que el número de pruebas estaba dentro de lo usual. Para cada prueba se modificó la cantidad de capas ocultas de la red (1 ó 2), el número de neuronas ocultas de cada capa (desde 48 hasta 96), la razón de aprendizaje de la red (desde 0,05 a 0,4, para un rango de 0 a 1), el momentum de la red (desde 0 a 0,6) y el número de ciclos de entrenamiento (de 1000 a 20000).

En particular, para el caso de los valores de la razón de aprendizaje, se siguió las recomendaciones de Elman, [2] con respecto a utilizar valores bajos de este parámetro para prevenir falta de generalización de la red ya entrenada.

La configuración de red más simple corresponde a la con menor número de capas y neuronas ocultas, la más completa a la con mayor número de capas y neuronas. Entre los extremos anteriores se sitúan las 44 pruebas restantes.

Así, del conjunto de pruebas realizados se seleccionó la prueba 12 como el de mejor desempeño, dado que fue la que mejor porcentaje de aciertos de validación obtuvo (82,96 %). Sin embargo, no fue la del menor error: 0,00011476 versus 1,349E-05 de la prueba 14, pero donde el error era menor, se perdía generalización lo que se reflejó en un menor porcentaje de validación. Así, la arquitectura de red elegida es la de una red Perceptron multicapa de 1 capa oculta y 72 neuronas en dicha capa.

El comportamiento del error de entrenamiento y el porcentaje de validación se ilustran en las figuras 15 y 16 respectivamente. Ambas se entregan en escala logarítmica.

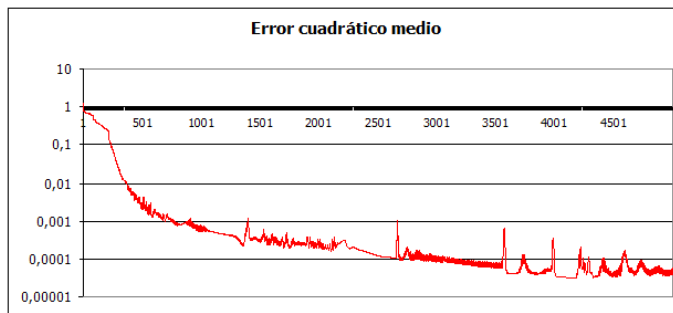


Figura 15: Gráfico de error de entrenamiento.

De la figura 15 se puede apreciar la convergencia del error hasta alcanzar un valor estable alrededor de 0,00011476, se observa que la convergencia es pausada y no brusca, esto se debe a una baja razón de aprendizaje, lo que garantizó la estabilización del error al final del bucle.

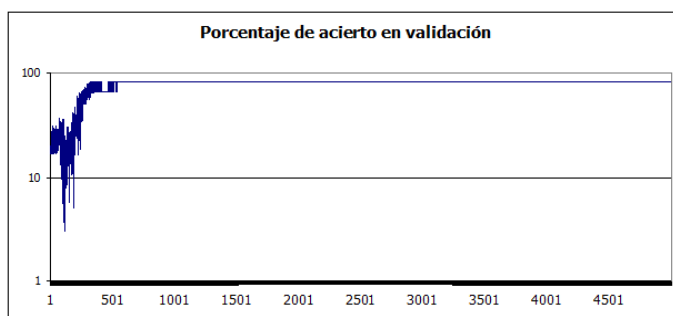


Figura 16: Gráfico del porcentaje de acierto en validación.

A diferencia del error de entrenamiento, el porcentaje de validación converge antes, aproximadamente en el ciclo 700, tal y como se observa en la figura 15. Sin embargo, el entrenamiento debe esperar que el error convergiera adecuadamente alrededor de un valor.

## 4.2. Resultados del Desempeño del Sistema Computacional

El estudio se llevó a cabo ocupando un computador personal con una CPU Intel Pentium dual-core de 1,73 GHz, memoria de trabajo RAM 1Gb y 60 Gb de disco duro, configuración que corresponde a un equipo de escritorio de una gama media.

En el caso del entrenamiento de la red, el tiempo de CPU utilizado para las pruebas varió dentro del rango de 56 segundos hasta 1 hora con 19 minutos y 56 segundos, según la complejidad de la topología de la red, correspondiendo el menor tiempo a una red simple y el mayor tiempo a la red más compleja. La carga de la CPU estuvo bajo el 70 % para todas las pruebas y, de igual forma, la memoria de trabajo no pasó el umbral de los 25 MB.

En el escenario de la ejecución de la red ya entrenada para clasificar una entrada dada, los recursos fueron notoriamente menores, siendo el tiempo de CPU menor a un minuto, la carga de CPU menor a un 30 % y la memoria de trabajo menor a 5 MB en todas las ejecuciones.

Por lo tanto, un computador de escritorio de gama media entrega la plataforma de hardware necesaria para una correcta ejecución del sistema computacional de la red neuronal.

## 4.3. Resultados del Problema en Estudio

### 4.3.1. Caso Datos en Bruto

En general, la red identificó adecuadamente un 83,33 % del total de entradas. El desglose se ilustra en la figura 17, donde se muestra el número de patrones identificados según el tipo ingresado. Siendo el mejor desempeño para

el patrón normal (91,23 % de efectividad) y el peor en el caso de un patrón de “Escalón positivo” (0 %).

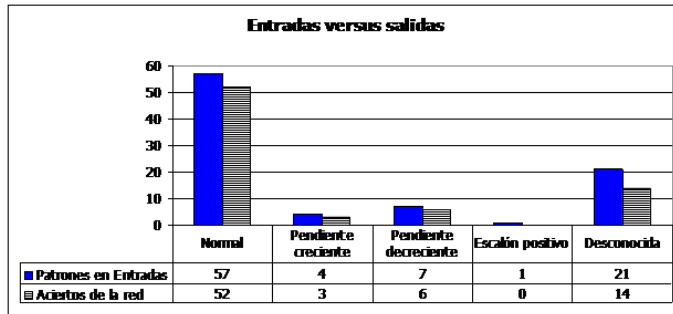


Figura 17: Desglose de patrones junto con clasificaciones realizadas por la red.

En los diferentes tipos de patrón conocidos, menos el escalón, se obtuvo un 66 % de acierto como mínimo.

En particular, se tiene que un 23,33 % del total de entradas corresponde a patrones no definidos ni conocidos. Se observa que las formas de las curvas son compuestas entre varios patrones a lo que debe sumarse una alta dispersión de los datos en algunos casos y, en general, ruido en los datos.

Dentro de los patrones conocidos existe el de “Escalón negativo” pero que no se hizo presente en el conjunto de datos analizados y la red tampoco lo identificó erróneamente.

En la siguiente matriz de cuatro cuadrantes (tabla 2) se ilustra los porcentajes en que la red identificó correctamente los patrones de entrada cuando eran conocidos e incorrectamente los mismos. Además, por otro lado entrega los porcentajes en que la red no identificó correctamente los patrones de entrada conocidos y los desconocidos.

	Identificación correcta de la red	Identificación incorrecta de la red
Entrada de Patrón conocido	<b>88,40 %</b>	<b>11,6 %</b>
Entrada de Patrón desconocido	<b>66,66 %</b>	<b>33,3 %</b>

Tabla 2: Eficiencia de la red en la clasificación de patrones.

De la tabla 2 se aprecia que la red obtiene una eficiencia de un 88,40 % para el caso de patrones conocidos, lo cual es un valor bueno. La eficiencia bajó ostensiblemente para los patrones desconocidos, en el fondo, no es que la red los identifique sino que no computa una solución de un patrón conocido en esta situación.

### 4.3.2. Caso Datos Filtrados

En general, la red identificó adecuadamente un 70 % del total de entradas. El desglose se ilustra en la figura 18, donde se muestra el número de patrones identificados según el tipo ingresado. Siendo el mejor desempeño para el patrón de “Pendiente creciente” (100% de efectividad) y el peor en el caso de un patrón de “Escalón positivo” (0%).

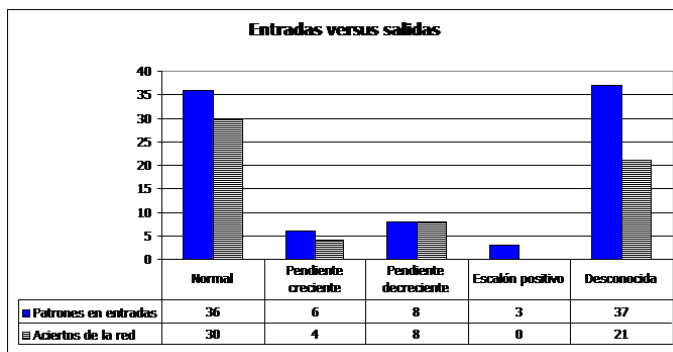


Figura 18: Desglose de patrones junto con clasificaciones hechas por la red.

De la figura 18 se puede apreciar que para los patrones conocidos, exceptuando el caso del “Escalón positivo”, la red tuvo un buen rendimiento en la clasificación de éstos.

En particular, se tiene que un 41,22% del total de entradas corresponde a patrones no definidos ni conocidos. Se ve que las formas de las curvas son compuestas entre varios patrones a lo que debe sumarse una alta dispersión de los datos en algunos casos.

Es de mencionar que dentro de los patrones conocidos existe el de “Escalón negativo” pero que no se hizo presente en el conjunto de datos analizados y la red tampoco lo identificó erróneamente.

En la siguiente matriz de cuatro cuadrantes (tabla 3) se ilustra los porcentajes en que la red identificó correctamente los patrones de entrada cuando eran conocidos e incorrectamente los mismos. Además, por otro lado entrega los porcentajes en que la red no identificó correctamente los patrones de entrada conocidos y los desconocidos.

	Identificación correcta de la red	Identificación incorrecta de la red
Entrada de Patrón conocido	<b>83,33 %</b>	<b>16,67 %</b>
Entrada de Patrón desconocido	<b>55,26 %</b>	<b>44,74 %</b>

Tabla 3: Porcentajes de acierto en clasificación.



De la tabla 3 se puede apreciar que la red obtiene una eficiencia de un 83,33% para el caso de patrones conocidos, lo cual es un valor muy apropiado bueno. La eficiencia baja considerablemente para los patrones desconocidos; en el fondo, no es que la red los identifique sino que no computa una solución de un patrón conocido en esta situación.

---

## 5. Conclusiones y trabajo a futuro

---

### 5.1. Conclusiones

Con respecto a la clasificación de la salida del horno por medio de la herramienta computacional, se encontró que el porcentaje de eficiencia fue de un 83,33%, que corresponde a un porcentaje aceptable y, además, es consistente con lo obtenido en la fase de entrenamiento (82, 96%) de la red neuronal contenida en el programa. Dado que los datos eran de un proceso real, lo que implica una componente de ruido inherente en los valores, se volvieron a procesar los datos pero con un filtrado previo, obteniendo un rendimiento similar para patrones conocidos, lo que indica que la red puede de trabajar con datos en crudo.

Además, al analizar los resultados, se encontraron falencias en el proceso de entrenamiento, siendo éstas: definición no adecuada del patrón “escalón” puesto que el cambio de valor objetivo es gradual en la realidad y no directo como en los datos de entrenamiento, falta de otros patrones típicos como cíclico por ejemplo y flexibilizar el esquema de ventana de muestreo fija a uno dinámico, pues en algunas ventanas fijas se mezclan el inicio y el fin de un patrón conocido, lo cual inducía a un error de clasificación por parte de la red.

En cuanto al campo de aplicación de esta metodología, es aplicable para evaluar el funcionamiento del control automático de cualquier proceso continuo o semi continuo. Además, es aplicable también en el ámbito del control de calidad para buscar patrones en cartas de control, por ejemplo, en las cartas I individual se puede aplicar directamente.

Acerca del uso de la metodología, se puede comentar que una vez entrenada la red, la utilización de la herramienta de clasificación es directa sin la necesidad de manejar conceptos de redes neuronales.

En lo relativo a las dificultades o requerimientos que se presentaron, se puede decir que el principal costo para la aplicación de la red es la necesidad de una gran cantidad de datos conocidos de entradas y salidas para poder entrenarla adecuadamente, por otro lado, en cuanto a las competencias requeridas para la interpretación de resultados se requiere un conocimiento de la operación del proceso bajo evaluación para poder explicar las causas a los patrones “anormales” que puedan encontrarse.

## 5.2. Trabajos futuros

En cuanto a la topología de la red, dado que sólo se trabajó con una arquitectura Perceptron multicapa, se propone probar con arquitecturas de red de base radial y de neuronas recurrentes para contrastar el porcentaje de efectividad y los tiempos de entrenamiento de cada caso.

Con respecto al conjunto de patrones, se propone incrementar el número de elementos, con otros tipos de patrones de origen teórico y/o práctico, para cubrir un amplio rango de patrones en señales de terreno.

Cambiar el sistema computacional a un esquema de una ventana de muestreo de tamaño fijo y estática en el tiempo por uno de una ventana dinámica en el tiempo, que se realice el muestreo de la señal de salida del proceso en tiempo real, de modo de abolir la mezcla de patrones conocidos debidos a una ventana fija de tiempo.

Se puede reorientar fácilmente el sistema computacional desarrollado para ser utilizado en la simulación de procesos industriales, lo que permite evaluar los procesos en cuanto a rendimiento y en predicción de las salidas de estos para entradas de interés.

**Agradecimientos:** a Mónica Meneses por su apoyo y más.

## Referencias

- [1] Chesnokov, Y. Backpropagation Artificial Neural Network in C++. <http://www.codeproject.com>., 2008.
- [2] Elman, J. Learning and development in neural networks: the importance of starting small. *University of California*., 1992.
- [3] Fu, T., Chung, F., Ng, V., Luk, R. Pattern Discovery from Stock Time Series Using Self-Organizing Maps. *Department of Computing Hong Kong, Polytechnic University*, 2001.
- [4] Gutiérrez, H., Salazar, R. Control estadístico de calidad y seis sigma. *Mc Graw Hill*, 2005.
- [5] Guh, R., Shiue, Y. Effective Pattern Recognition of Control Charts using a Dynamically Trained Learning Vector Quantization Network. *Journal of the Chinese Institute of Industrial Engineers*, Vol. 25, Nro. 1, 2008.
- [6] Hann, F., Kostanic., I. Principles of neurocomputing for science & engineering. *Mc Graw Hill*, 2000.
- [7] Kirillov, A. Aforge.Net Framework. <http://code.google.com/p/aforge/>, 2008.

- [8] Madhusudanan, O. Brainnet 1 - A Neural Network Project - With Illustration And Code - Learn Neural Network Programming step By Step And develop a simple Handwriting Detection System. <http://www.codeproject.com>, 2006.
- [9] Madhusudanan, O. NXML - Introducing an XML Based Language To Perform Neural Network Processing, Image Analysis, Pattern Detection, Etc. <http://www.codeproject.com>, 2006.
- [10] Matich, D. Redes Neuronales: Conceptos básicos y aplicaciones. 2001.
- [11] Microsoft, MSDN Library. 2003.
- [12] Microsoft, Introduction to Visual Basic 2005. 2005.
- [13] Muñoz, A. Aplicación de Técnicas de Redes Neuronales Artificiales al Diagnóstico de Procesos Industriales. *Universidad Pontificia comillas Madrid*, 1996.
- [14] Ogata, K. Ingeniería de Control Moderna. *Prendice Hall*, Segunda Edición, 1993.
- [15] Olmedo, E., Valderas, J. Utilización de de Redes Neuronales en la Caracterización, Modelización y Predicción de Series Temporales Económicas en un Entorno Complejo. 2004.
- [16] Ruiz, C. Sistemas inteligentes en la Industria de Procesos: Experiencia, Actualidad y Perspectivas. *Argentine Symposium on Artificial Intelligence*, 2001.
- [17] Sagioglu, S., Besdok, E., Erler, M. Control Chart Pattern Recognition Using Artificial Neural Networks. *Turk J Elec Engin*, Vol.8, No.2., 2001.
- [18] Singh, S., Fieldsend, J. Pattern Matching and Neural Networks based Hybrid Forecasting System. *PANN Research*, Department of Computer Science, University of Exeter, UK, 2001.
- [19] Silva, R., Cruz, A., Hokka, C., Giordano, R. A Hybrid Feedforward Neural Network Model for the Cephalosporin C Production Process. *PANN Research*, Department of Computer Science, University of Exeter, UK, 2001.
- [20] Soto, L. Tutorial Visual C# 2005. <http://www.programacionfacil.com>, 2007.
- [21] Tham, M. Dealing with measurement noise. <http://lorien.ncl.ac.uk/ming/filter/filter.htm>, 1998.

- [22] Viñuela, P., Galván, I. *Redes de Neuronas Artificiales, Un enfoque Práctico*. *Prentice Hall*, 2004.
- [23] Winston, W. *Investigación de operaciones: Aplicaciones y algoritmos*. *Thomson*, 2004.