

# Un enfoque de redes neuronales hacia el control de calidad para procesos multivariados basados en estimación de densidades

Carlos D. Paternina A.\*, Eduardo Lerín V.\*\*, Erika V. Márceles G.\*\*\*  
Laboratorio de Robótica y Automatización de la Producción  
Universidad del Norte

## Resumen

*En este artículo se presenta un procedimiento no-paramétrico para identificación de no-conformidades en el control de procesos multivariados. El procedimiento propuesto, calcula una medición de la densidad para cada celda en un espacio particionado definido por el set de entrenamiento. Una vez la densidad para cada celda es calculada, se desarrolla una aproximación a una función continua de densidad utilizando redes neuronales de Retropropagación. El entrenamiento de la red neuronal sólo requiere información del proceso en-control, lo que significa que no se utilizan ejemplos negativos. La información de entrenamiento es pre-procesada para sopesar la carencia de información de entrenamiento fuera-de-control. Nuevos casos provenientes del proceso analizado son examinados bien sea que pertenezcan a la distribución en-control o fuera-de-control. La clasificación de las categorías: en-control y fuera-de-control, se fundamenta en la función de densidad de probabilidad aproximada. Los resultados de las simulaciones son utilizados para realizar comparaciones con otras tecnologías existentes.*

**Palabras claves:** Redes neuronales, entrenamiento, retropropagación, control, clasificación.

## Abstract

*In this project a nonparametric procedure for identification of errands in multivariate process control is presented. The proposed procedure calculates a density measure for each cell in a partitioned space defined by the training data set. Once the density for each cell is calculated, a continuous density function is approximated using backpropagation neural networks. Training only requires data from the in-control process which means no negative examples are*

Fecha de recepción: 14 de febrero de 2001

\* Coordinador Laboratorio de Robótica y Automatización, Universidad del Norte. Ingeniero Industrial, Universidad del Norte. Maestría en Ingeniería Industrial, University of South Florida. Ph.D. en Ingeniería Industrial de esta última universidad. Miembro de IIE (Institute of Industrial Engineers), SME (Society for Manufacturing Engineers) y de RI (Robotics International) de SME. [cpaterni@uninorte.edu.co](mailto:cpaterni@uninorte.edu.co)

\*\* Ingeniero Industrial, Universidad de Houston. Maestría en Ingeniería Industrial, University of South Florida. Ph.D. en Ingeniería Industrial de University of South Florida. Miembro de IIE (Institute of Industrial Engineers). [lerin@eng.usf.edu](mailto:lerin@eng.usf.edu)

\*\*\* Estudiante de décimo semestre del programa de Ingeniería Electrónica de la Universidad del Norte. Miembro del Grupo de Automatización y Robótica. [emarcele@unimail.uninorte.edu.co](mailto:emarcele@unimail.uninorte.edu.co)

utilized. Training data is pre-processed to overcome the lack of out-of-control training set. New points coming from the analyzed process are tested if they belong either to the in-control or out-of-control distribution. Classification for in- or-out-of-control is done based on the approximated density probability function. Results from simulated data are used to compare with existing methodologies.

**Key words:** Neural networks, training, Backpropagation, control, clasification.

---

## INTRODUCCIÓN

Las cartas de control son utilizadas para detectar si un proceso está dentro de control o no. Es decir, ellas determinan si la distribución subyacente de un proceso, o sus parámetros, tales como el promedio y la dispersión, permanecen a un nivel deseado. También son una buena alternativa para distinguir entre la variabilidad natural de los procesos (por ejemplo, variaciones en el proceso que no son económicamente viables de identificar o corregir) y la variabilidad asignable.

La suposición de normalidad es razonable, especialmente si la metodología del control está fundamentada en el simple promedio de  $n$  observaciones que son tomadas en un punto específico del procedimiento de control. De esta forma, mediante el Teorema del Límite Central se asevera que los valores medios del proceso tienen una distribución aproximadamente normal. Sin embargo, algunas veces, debido a razones económicas o técnicas, puede no ser posible recolectar un tamaño muestra  $n$  lo suficientemente grande, y sólo una observación al tiempo puede ser tomada. De esta forma, se necesita construir cartas con observaciones individuales. En tales casos, el Teorema

del Límite Central y, en consecuencia, la suposición de normalidad dejan de ser confiables.

La clasificación es la base de los procedimientos de control no-paramétricos. La discriminación se basa en las dicotomías de las características. En este estudio se discuten tanto estados en-control como fuera-de-control de un proceso según sus características. Para construir una regla de reconocimiento de patrones se necesita conocimiento *a priori* del proceso. Este conocimiento *a priori* es denominado *set de entrenamiento*. Entre mayor es la cantidad de información disponible sobre el proceso, la respuesta derivada de la regla de decisión llega a ser más precisa.

Formalmente, un clasificador puede ser visto como un mapeo:  $X: \Omega \rightarrow \mathbb{R}^d$ ,  $Y: \Omega \rightarrow \{0,1\}$ , donde  $X$  es el vector de muestra del proceso y  $Y$  es la correspondiente respuesta esperada. El error propio de un clasificador está usualmente definido por el número (o promedio) de muestras erróneamente clasificadas.

Se ha diseñado un clasificador que toma las muestras del proceso como entradas y proporciona un valor de densidad que indica a cuál clase la muestra se acomoda mejor, bien sea en-control o

fuera-de-control; el resultado es un procedimiento libre de distribución, multivariado, para control de calidad.

En este estudio las comparaciones se han limitado al espacio bi-dimensional para efectos de simplicidad, y se asume que lo mismo aplica a dimensiones mayores.

### 1. MOTIVACIÓN SOBRE REDES NEURONALES

La metodología de este trabajo se basa en la estimación no-paramétrica de la densidad de unas determinadas celdas, de acuerdo a un set de entrenamiento en-control. El espacio de los datos se particiona [1], y se asigna una probabilidad de ocurrencia (frecuencia) de visita a cada celda. La clasificación de las muestras, basada en reglas de partición, consiste en dividir el espacio de datos

$\mathcal{R}^d$  (donde  $d$  es el número de características que se debe controlar) y distribuir la nube de datos entre las diferentes celdas en el espacio. La reglas de partición, a pesar de que son efectivas, están restringidas a dar información discreta, lo que significa que la respuesta que se obtiene de las celdas en el espacio particionado puede llevarnos a clasificaciones erróneas debido a la aleatoriedad. La figura 1 ilustra el problema con dos celdas adyacentes con información contradictoria. Esta situación motiva a aproximar una función continua con los datos obtenidos del espacio particionado. Las redes neuronales son consideradas aproximadores robustos de funciones no-paramétricas que hacen innecesaria cualquiera suposición sobre la distribución de los datos. Más aún, cuando se manejan datos multivariados, es casi imposible hacer alguna suposición sobre la distribución.

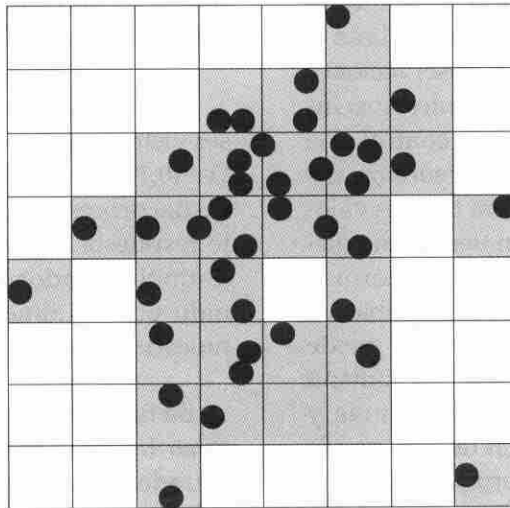


Figura 1. Información ambigua en el espacio particionado

## 2. PRE-PROCESAMIENTO DE LOS DATOS

El primer experimento de esta investigación consistió en el entrenamiento de una red neuronal con un set de entrenamiento de 500 datos en la condición en-control (ver Sección 6). Como es obvio, por ser la red entrenada con un solo tipo de datos, la función de aproximación generada por la red era capaz de hacer asociaciones perfectas con los datos en-control de prueba (set de prueba). Pero, al probar la red con datos fuera-de-control, el error obtenido fue considerable. La red hizo una generalización únicamente para los datos en-control. De esta forma, se hizo necesario un refinamiento en el experimento, con el fin de obtener una buena estimación de la función de densidad.

El concepto de particionar el espacio de datos puede ser aprovechado. Los datos en-control, cuando son analizados por medio de particiones, aumentan el espacio de acuerdo con los puntos más extremos en la nube de datos. Esto, a su vez, producirá celdas, en la partición, sin ningún tipo de información. La carencia de información sobre los datos conlleva a una situación fuera-de-control, permitiendo de esta manera que la red generalice sobre ambos los datos en y fuera-de control.

### 2.1. Reglas de Partición

La clasificación de muestras basada en reglas de partición consiste en dividir el espacio de datos  $\mathfrak{R}^d$  (donde  $d$  es el número

de características o variables en consideración) en celdas disyuntas para las cuales la decisión se hace de acuerdo con la mayoría de votos entre los valores de respuesta de las muestras del set de entrenamiento que caen en la celda. Formalmente, una regla de partición se define de la siguiente forma:

Sea  $D_m = \{(x_1, y_1), \dots, (x_m, y_m)\}$  el set de entrenamiento. Si  $x$  representa un punto nuevo y  $A(x)$  representa la partición en la que  $x$  reside, la clasificación  $g_m(x)$  del nuevo punto  $x$  está dada por:

$$g_m(x) = \left\{ \begin{array}{l} 0, \text{ si } \sum_{i=1}^m I_{\{Y_i=1\}} I_{\{x_i \in A(x)\}} \leq \sum_{i=1}^m I_{\{Y_i=0\}} I_{\{x_i \in A(x)\}} \\ 1, \text{ o/w} \end{array} \right\}$$

donde  $I_{\{\cdot\}}$  es la función indicador.

Un aproximación a la partición muy útil es la Regla del Histograma Cúbico (CHR, *Cubic Histogram Rule*), en la que el espacio muestral es equitativamente dividido en hipercubos  $P_m = \{A_{m1}, A_{m2}, \dots\}$  de volumen  $h_m^d > 0$ . Para cada  $x \in \mathfrak{R}^d$ ,  $A(x) = A_{mi}$  si  $x \in A_{mi}$ .

### 2.2. Regla del Histograma de Densidad

Este enfoque es completamente no-paramétrico, lo que permite al usuario detectar no conformidades sobre el valor promedio de un proceso. Tiene la ventaja de trabajar sólo con datos en-control, contrariamente a cualquier función de discriminación, que requiere un set de entrenamiento de datos en-control y fuera-de-control.

Los siguientes pasos ilustran el algoritmo utilizado en esta investigación para controlar muestras provenientes del proceso monitoreado:

Para un set de entrenamiento de datos en-control  $D_m$  dado,

1. Considere una partición  $P_m = \{A_{m1}, A_{m2}, \dots\}$  en  $\mathcal{R}^d$ , de tamaño  $h_m > 0$ , del espacio que encierra  $D_m$ .
2. Encuentre el punto central  $c_i$  de cada partición  $A_{mi}$  de volumen  $h_m^d$ .
3. Calcule:

$$dG_m(c_i) = \frac{1}{m h_m^d} \sum_{j=1}^m I_{\{(x_j) \in A_{mj}\}}$$

4. Sea  $x$  un nuevo dato a ser clasificado y  $A_{mk}$  el contenido de la celda del nuevo dato.
5. La estadística de control para el punto  $x$  es  $dG_m(c_k)$ .
6. Grafique la estadística en un cuadro con base de tiempo, y continúe monitoreando el proceso.

Siendo que  $0 \leq dG_m(x) \leq 1$  y que la estadística es del tipo "entre más alto, mejor", el cuadro de control necesita un límite de control menor  $0 < L < 1$  tal que, cuando  $dG_m < L$ , se catalogue al proceso como fuera-de-control. Claramente, el error de clasificación DHR II tiene un umbral más bajo en la Pérdida óptima de Bayer sin tendencia a la convergencia. Si el número de celdas en la partición aumenta, un mayor número de celdas no tendrá información sobre densidad y, en consecuencia, el error tipo I se incrementará, a su vez que el error tipo

II disminuirá y eventualmente se hará cero cuando el tamaño de la celda se haga cero.

### 3. RETROPROPAGACIÓN

El algoritmo de Retropropagación es una aproximación de primer orden de la técnica de descenso más profundo [2], y por lo tanto es aleatorio por naturaleza. La aleatoriedad proporciona una tasa de convergencia lenta para el algoritmo; debido a que para este estudio la rapidez del entrenamiento es irrelevante (el máximo tiempo de entrenamiento fue de aproximadamente 2.5 horas), se puede afirmar que el algoritmo trabaja bien para nuestro problema. Como y se mencionó, el propósito de usar el algoritmo de Retropropagación es aproximar una función continua con la información obtenida de la partición (ver sección 2). El centro de cada celda es ingresado a la red neuronal como el patrón de entrada para el entrenamiento, siendo el valor objetivo de cada entrada el valor de densidad asignado a la celda. Este método, además de permitir el manejo del número de datos de entrada (mediante la variación del número de particiones en el espacio de datos), proporciona a la red los puntos extremos necesarios para un entrenamiento adecuado.

El algoritmo de Retropropagación (*Backpropagation*) es uno de los más utilizados en las aplicaciones de redes neuronales. Es un método de entrenamiento general para redes *feedforward* de varias capas. Utiliza neuronas que son no-

lineales pero diferenciables. La función de transferencia utilizada en las neuronas es la función sigmoidea, la cual limita el rango del resultado entre 0 y 1. El proceso de entrenamiento consta de dos pasos: un paso hacia adelante y un paso hacia atrás. En el paso hacia adelante, las entradas son aplicadas a la red y las salidas son calculadas. En el paso hacia atrás, el error entre la salida obtenida y la deseada es calculado, para ajustar los pesos de la capa de salida. Debido a que no se tiene valores objetivos para las capas escondidas, el error calculado se propaga hacia dichas capas, de manera que los pesos de las capas escondidas son también ajustados. Se dice que el algoritmo de Retropropagación es un «método de gradiente moderado», ya que los pesos son ajustados de forma tal que el cambio del error con respecto al cambio de los pesos es negativo para cada peso. De esta forma, esta técnica de entrenamiento trata de minimizar los cuadrados de las diferencias entre los valores obtenidos y deseados, a través de todo el set de entrenamiento.

Entre las ventajas de este algoritmo se encuentra su aplicabilidad en una amplia gama de problemas y su sólida fundamentación matemática. Sus desventajas son la posibilidad de quedarse estancado en un mínimo local, la carencia de un método efectivo para la selección de la arquitectura de la red y de la constante de aprendizaje y, por último, los largos tiempos de entrenamiento requeridos.

### 3.1. Escalamiento de los datos

Con el fin de hacer el proceso de clasificación independiente de la magnitud de los datos, el escalamiento se hace indispensable. Para nuestro caso de distribución normal bivariada, el escalamiento no es necesario, debido a que los datos son simulados y directamente generados estandarizados (generación aleatoria normal bivariada estándar sin covarianza). Por otra parte, para el caso de variables aleatorias tipo Gamma, el escalamiento se hace necesario, ya que no existe una función estándar para la distribución gamma.

Además de escalar las entradas es necesario escalar las salidas al apropiado nivel de densidad. Esto es debido a que los valores de densidad para el espacio particionado varían desde cero hasta un valor realmente pequeño (no es una probabilidad acumulativa). Esto hace que el algoritmo aproxime todos los valores a cero, clasificando incorrectamente los datos de prueba en-control.

### 3.2. Arquitectura y Manejo de la Red

El manejo de la arquitectura de la red es cuestión de prueba y error. Inicialmente se comenzó a trabajar con un tipo de redes con una capa escondida y se hicieron pruebas con los datos (set de prueba) para cada una de ellas. Luego se modificó el número de neuronas escondidas desde 3 hasta 8 neuronas obteniéndose el mejor error de clasificación para todo el conjunto en la configuración 2-6-1 (2 entradas, 6 neuronas escondidas y 1

salida). La comparación fue fundamentada en el error de clasificación y no en la suma de los cuadrados de los errores que se obtienen del *software*, puesto que lo que se desea es que se resuelva el problema de clasificación.

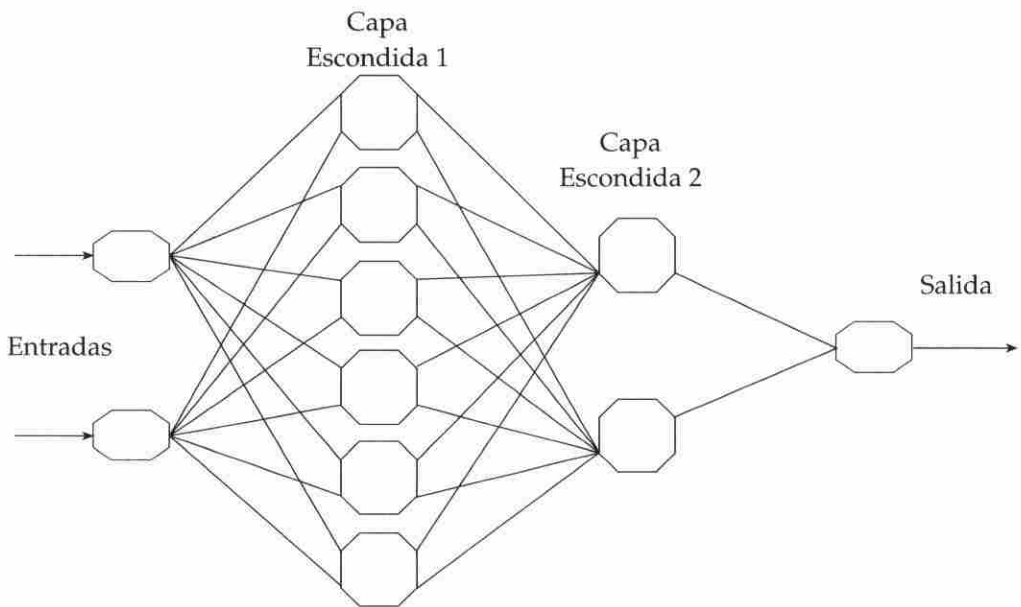
En un segundo intento se hizo una implementación de una red con dos capas escondidas y se modificó el número de neuronas en ambas capas escondidas. En este caso, la mejor configuración obtenida fue la de una red completamente interconectada 2-6-2-1 (2 entradas, 6 neuronas en la primera capa escondida, 2 neuronas en la segunda capa escondida y una salida), basada también en el error de clasificación (ver figura 2). Es todavía más interesante que para los dos ejemplos utilizados

(bivariada normal y bivariada gamma) la misma configuración dio el mejor resultado de clasificación en cada caso.

La unidad computacional escogida fue la función sigmoideal por defecto (*logsig*) con salidas transformadas en el rango (0,1) que asignan de forma apropiada un valor de densidad de probabilidad luego de volver a escalar las salidas.

#### 4. MEDIDAS DE RENDIMIENTO

El rendimiento del procedimiento de control con la metodología propuesta puede ser medido en términos del error de clasificación obtenido con el set de prueba en-control y fuera-de-control (referenciado únicamente a un set de



**Figura 2.** Red de Retropropagación con estructura 2-6-2-1

entrenamiento con datos en-control). Un error de clasificación combinado se calcula como la pérdida total ( $L_{ANN}$ ), que puede ser comparado con la frontera más baja dada por otros métodos (por ejemplo, Data depth, profundidad de data).

## 5. EJEMPLOS NUMÉRICOS

Se utilizaron simulaciones para generar dos diferentes sets de entrenamientos para esta investigación. Inicialmente se generó aleatoriamente un set de datos normales bivariados, y luego se generó un set de datos con distribución gamma bivariada. El objetivo consistía en evaluar el procedimiento de clasificación con ambas distribuciones multivariadas simétrica y no-simétrica.

En ambos casos fueron generados 540 datos en-control, de los cuales sólo 500 fueron utilizados como set de entre-

namiento. Los 40 restantes, junto a otros 40 datos fuera-de-control, fueron utilizados para constituir el set de prueba. El valor de umbral (límite de control L) se escogió  $L = 1/500$ . Las comparaciones se hicieron con referencia al bien-conocido procedimiento no-paramétrico basado en medidas de la profundidad de los datos (data depth) [3].

## 6. RESULTADOS

La tabla muestra los resultados para ambos conjuntos de datos, con distribución normal bivariada y con distribución gamma bivariada. La primera columna de la tabla muestra el nivel de partición utilizado para el manejo del número de datos en el set de entrenamiento. Un valor de  $p = 10$  significa que se hicieron 10 particiones en cada variable obteniéndose, de esta forma, 100 celdas. Es decir, la red fue alimentada con 100 parejas de datos de entrenamiento (en-

Error de clasificación para los sets de prueba con distribución a) *Normal bivariada* y b) *Gamma bivariada*

Particiones p	DHR error	ANN error
10	15	15
15	12	13
20	12	11
25	14	11
30	16	12
35	22	15
40	25	17
45	24	23
50	26	23

Error de profundidad de Mahalanobis = 11

Particiones p	DHR error	ANN error
10	25	24
15	24	22
20	28	22
25	26	21
30	23	20
35	28	18
40	25	19
45	31	25
50	30	28

Error de profundidad de Mahalanobis = 27



trada-valor objetivo). La segunda columna muestra el error de clasificación discreta mediante la regla del histograma cúbico en cada nivel de partición. Finalmente, la tercera columna muestra los resultados obtenidos con el enfoque de la red neuronal.

El error tanto para el enfoque de la regla del histograma de densidad (DHR) como para el de red neuronal artificial (ANN), está dado por el número total de puntos erróneamente clasificados, correspondiente al set de prueba de 80 datos en- y fuera-de-control. En la tabla 1 se puede observar que para datos distribuidos simétricamente (normal bivariada), la red neuronal produce un error igual al que se obtiene con el método de profundidad de datos de Mahalanobis para los casos en los que  $p$  es 20 y 25. Para otros números de particiones, la red neuronal no soluciona el problema como se desea. Por otra parte, para datos con distribución no-simétrica (gamma bivariada), la red neuronal produce un error mejor que el obtenido con el método de Mahalanobis para todos los niveles de partición analizados menos uno. El método de Mahalanobis es un buen punto de referencia para hacer la comparación, debido a que es el clasificador más eficiente, para ese tipo de problemas, encontrado en la literatura [3,4].

### CONCLUSIONES

Las redes neuronales proporcionan un enfoque muy útil para aproximar funciones continuas de densidad corres-

pondientes a un set de datos sin ninguna suposición sobre su distribución, lo que significa que es un procedimiento completamente no-paramétrico. La idea de particionar el espacio de los datos simplifica de gran manera el procedimiento de clasificación de los datos en tiempo real sin incrementar significativamente el error para un tamaño adecuado de partición. Este tamaño de partición puede ser refinado empíricamente, de acuerdo con el error de clasificación obtenido.

En este estudio se tuvo problemas con los valores objetivos. Antes de escalarlos se pensó que debido a que ellos eran valores de densidad, la red iba a entregar directamente un valor aproximado de densidad. Luego de muchos intentos con diferentes números de particiones pudimos notar que la red estaba aproximando todos los datos a cero, entregando respuestas erróneas a nuestro problema de clasificación. Se pensó que el escalamiento de la salida era una forma de aproximar los valores originales de densidad a valores de probabilidad acumulativa, lo que puede ser directamente comparado con el nivel de importancia de la prueba de hipótesis (dado el umbral al límite de control).

### Referencias

- [1] DEVROYE, L., GYÖRFI, L., y LUGOSI, G. A *probabilistic Theory of Pattern Recognition*. New York, Springer-Verlag, 1996.
- [2] HAYKIN, Simon. *Neural Networks. A Comprehensive Foundation*. New Jersey, Mac-Millan College Publishing Company, 1994.

- [3] LIU, R.Y. «Control Charts for Multivariate Processes». *Journal of the American Statistical Association*. December, 1995, vol. 90, N° 432, Theory and methods.
- [4] LIU, R.Y., y SINGH, K. «A Quality index Based on Data Depth and Multivariate Rank Tests». *Journal of the American Statistical Association*, March, 1993, vol. 88, N° 421, Theory and Methods.

## Bibliografía

- BOSE, N.K., y LIANG, P. *Neural Network Fundamentals with Graphs, Algorithms and Applications*. McGraw-Hill series in Electrical and Computing Engineering.
- GLICK, N. «Sample-based Multinomial Classification». *Biometrics*. Vol. 29, p. 241-256, 1973.