

# Controlador neuroborroso: Un caso de estudio

GUTIERREZ DE MESA, J.A.\*; RODRIGO YÁNES, J.A.\*\*; GONZALEZ SOTOS, L.\*; GARCIA BARRIOCANAL, E.\*; SICILIA URBÁN, M.A.\*\*\*; MUEL MUEL, E. y BARCHINO PLATA, R.\*;

\*Departamento de Ciencias de la Computación (Universidad de Alcalá) [jantonio.gutierrez, leon.gonzalez}@uah.es]

\*\*Departamento de Automática (Universidad de Alcalá) [jrodrigo@aut.uah.es]

\*\*\*Departamento de Ciencias de la Computación (Universidad Carlos III) [msicilia@inf.uc3m.es]

España

## Resumen

En el presente artículo se desarrolla la necesidad de procesar la información mediante una operación elemental de selección. La operación se efectúa sobre el producto escalar de un vector de información, recogida en tiempo real, con un vector de ponderaciones que deberá ser ajustable en el mismo tiempo. Las unidades empleadas estarán definidas en un conjunto de valores discretos y deberán ser comparadas, continuamente, con unos vectores de umbrales de referencias de cierta vaguedad. El tipo de operación, que se utilizará para desarrollar el controlador, viene definido mediante un proceso de inferencias sobre los conectivos lógico-borrosos denominados normas y conormas triangulares [1], que se emplean en un proyecto de control real en una central de encendido electrónico [2].

## 1. Introducción

El motivo de usar controladores *fuzzy* reside en que pueden implementarse en aquellas situaciones, en las que el tiempo de respuesta, debe de ser muy rápido a partir de las señales de entrada que cambian en un orden de tiempo de decenas de microsegundos. Por otra parte, las redes neuronales tienen la ventaja de poder ser entrenadas para responder a estímulos y ofrecer una función de salida muy rápida. Además, dependiendo de su programación, el entrenamiento que se debe realizar sobre la red puede hacerse sin supervisión, de tal forma que, una sonda preparara la función de pertenencia para la clasificación de los propios ejemplos según se van produciendo. El problema de la clasificación de las clases en las fronteras del universo a controlar se hace más patente cuando el grado de definición de los ejemplos es borroso.

En este sentido, nuestro controlador lo podemos considerar como un análisis formal de aprendizaje de un perceptrón nítido. Este perceptrón se aumenta con la utilización adicional de una función de pertenencia, ofrecida experimentalmente, y que permitirá separar las distintas clases, a fin de, resolver los conflictos obtenidos al suavizar las regiones frontera del universo a controlar [3]. Con estas consideraciones, la red neuronal aprende la conjunción de distintas premisas borrosas, de tal forma que, la conjunción de reglas antecedentes tiene que realizarse con una norma triangular diferenciable para procesar reglas de la forma “si  $x_1$  es  $B_{j1}$  y  $x_2$  es  $B_{k2}$ , entonces  $a$  es  $g_i$ ” siendo  $g_i$  un valor nítido del estado a controlar  $a$ .

Para poder proceder a evaluar la función que a cada entrada hace corresponder una salida según este tipo de tecnología, partiremos de un proceso de *fuzzificación*, entendido como el método para convertir valores continuos en valores discretos a los que, normalmente, se le asocian atributos lingüísticos de la forma mucho, poco, bastante, etc. Posteriormente, y utilizando la máquina de inferencia propuesta, se obtendrán unos valores que habrá que *defuzzificar* para obtener, nuevamente, valores precisos; al fin y al cabo se está estableciendo una función de entrada-salida de la forma  $g_i = \mathfrak{I}(x_1^i, x_2^i, \dots, x_n^i)$  con  $i$  variando en el tiempo.

Una vez modelado el controlador buscamos la función que mejor resultado pueda ofrecer en nuestro universo de discurso y, tras algunos ensayos, nos limitamos, en el presente estudio, a la utilización del controlador de Mamdani por ser uno de los más utilizados en el mundo empresarial. El controlador de Mamdani utiliza conjuntos borrosos tanto para la entrada como para la salida y, en algunos casos, como en el nuestro, necesita una defuzzificación para poder generar la función de entrada-salida.

## 2. Marco de actuación

La definición formal del controlador se establece de la siguiente forma: Sea  $X$  un espacio arbitrario de entradas posibles al controlador, sean  $A_1, A_2, \dots, A_n$  y  $B_1, B_2, \dots, B_n$  subconjuntos de las entradas  $X$ , en nuestro caso subconjuntos fuzzy normalizados de  $X$ , y de  $\mathfrak{R}^n$  con la función medible de Borel, respectivamente. Sea  $T$  la  $t$ -norma medible de Borel, y consideremos el sistema de reglas ( $i = 0, 1, 2, \dots, n$ ) “SI  $x$  es  $A_i$  ENTONCES  $u$  es  $B_i$ ”. Entonces el *controlador de Mandani* viene definido por la siguiente función de entrada-salida  $F_M : (x) \rightarrow \mathfrak{R}^m$ , donde

$$F_M(x) = \frac{\int_{\mathfrak{R}^m} \mu_R(x, u) u du}{\int_{\mathfrak{R}^m} \mu_R(x, u) du}$$

con  $\int_{\mathfrak{R}^m} \mu_R(x, u) du > 0$ , donde las funciones miembro

$\mu_R$  de la relación fuzzy  $R$  en  $X \times \mathfrak{R}^m$  viene dada por:

$$\mu_R(x, u) = \max[T(\mu_{A_1}(x), \mu_{B_1}(u)), \dots, T(\mu_{A_n}(x), \mu_{B_n}(u))]$$

Para que se satisfaga la fórmula anterior es necesario incluir la medida de borrosidad; sin embargo, en la mayor parte de las situaciones reales, esta necesidad viene cubierta por el modelo.

Con esta definición se introduce, de forma indirecta, un método de defuzzificación, llamado centro de gravedad, que utilizamos en el entrenador que a continuación proponemos.

La  $t$ -norma usada en el controlador es la  $T_M$ , también conocida como la de *máx-min-inferencia*.

### 3. La red neuronal

Las redes borrosas pueden aprender, a partir de ciertas fórmulas, de forma iterativa, lo que tienen que “aprender”. En nuestro caso particular, sea  $X = [x_1, x_2, \dots, x_n]$  la representación de un estado posible de las entradas y salidas que nuestro controlador debe de reconocer con situación positiva. Pongámonos en la situación de que una red calcula erróneamente este valor y le da un valor negativo. Esto significa que la red, utilizando unos pesos calculados por la función fuzzy anterior, le asigna unos pesos tales que  $w_1 x_1 + w_2 x_2 + \dots + w_n x_n < T$  en lugar de haber calculado  $w_1 x_1 + w_2 x_2 + \dots + w_n x_n \geq T$ . El error estará, necesariamente, en el cálculo de las ponderaciones, lo que nos llevará a introducir una pequeña modificación  $\delta$  en las ponderaciones de las señales de entrada con lo que, para cada paso de ajuste,  $w_1 x_1 + w_2 x_2 + \dots + w_n x_n$  se convertirá en

$$(w_1 + \delta x_1)x_1 + (w_2 + \delta x_2)x_2 + \dots + (w_n + \delta x_n)x_n$$

que, desarrollando, se ve que es mayor que la primer ecuación; deduciéndose que, en varias iteraciones, se logrará cambiar el signo de la desigualdad hasta alcanzar la clasificación correcta.

La ecuación frontera  $w_1 x_1 + w_2 x_2 + \dots + w_n x_n = T$  corresponde a un plano de  $n$  dimensiones que, de alcanzarse, indica que el sistema neuronal alcanza a clasificar elementos de dos clases, o en otras palabras, nos indica si las dos clases son separables. En algunas situaciones nos encontraremos con problemas en la situación frontera donde algunos elementos pudieran quedar mal clasificados.

En nuestro caso, como utilizamos controladores borrosos, el grado de pertenencia de los ejemplos de entrenamiento a las respectivas clases, para decidir sobre posibles soluciones al conflicto en la región de la frontera, nos fuerza a utilizar una variable  $\beta$  para señalar si el incremento de acercamiento continúa o ha terminado según una función de Heaviside:

$\mathfrak{I}(\mu_p(x_j) - \mu_A(x_j) + \mu_p(x_j)\mu_A(x_j) - 0,1)$ , donde  $\mu_p$  es el grado de pertenencia del elemento crítico con respecto a la clase propia y  $\mu_A$  a la clase ajena.

Siguiendo el caso de estudio, el algoritmo utilizado resuelve el problema del aprendizaje sin supervisión de la siguiente manera: El valor introducido en la corrección  $\delta$  será sustituido por  $\delta(|\mu_p(x_j) - \mu_A(x_j)|)$  lo que hace que cuanto más nos acerquemos a las fronteras de las clases de equivalencia más dudosa es su clasificación, sobre todo en un sistema de adiestramiento sin supervisión, en el que un sistema de reglas fuzzy intentan definir la satisfactibilidad de la fórmula, y, por tanto, menor será la modificación introducida en las ponderaciones.

De otra parte puede sustituirse la función escalón de Heaviside por algún tipo de función sigmoideal fácilmente evaluable para producir una mayor rapidez en el sistema de aprendizaje.

Con el sistema presentado, que a primera vista tiene la atractiva ventaja de mezclar las redes neuronales con lógica fuzzy, guiada por la capacidad de aprendizaje mediante el empleo de una batería de ejemplos junto a unos métodos de aplicar razonamiento vago, estamos aproximándonos a un modelo simplificado del razonamiento humano.

Representar funciones de pertenencia utilizando una red neuronal en sentido clásico es bastante sencillo, pero la determinación de las ponderaciones puede tener algunas complicaciones. En este sentido, si tratáramos de clasificar un universo a partir de dos neuronas, se puede precisar con exactitud el peso de los niveles sinópticos después de un número de pasos de ajustes. Sin embargo, si utilizamos una tercera neurona con capacidad de ajustar los parámetros de desplazamiento, mediante técnicas de retropropagación, podemos llegar a hacer que nuestra red se comporte rápidamente como un buen sistema experto, sobre todo si las anteriores neuronas se encuentran siempre excitadas y con ponderaciones producto de las anteriores según un tipo de conjunción definida mediante una norma triangular diferenciable (la implementación más sencilla sería efectuar el producto de las antecedentes).

Generalizamos el caso al estudio de  $n$  neuronas por capa y  $m$  capas con retropropagación para llegar a poder expresar nuestro universo de reglas y aplicar el algoritmo de aprendizaje cuya convergencia sea eficiente y no muy sensible al problema de “los mínimos relativos”. En nuestro caso empleamos un pequeño refinamiento del algoritmo de las Gradientes Conjugadas [4] debido a Myller en 1990.

### 4. Otras consideraciones de borrosidad

La red descrita anteriormente parte de unos valores definidos como operadores lingüísticos con un cierto grado de vaguedad pero determinado, normalmente, por una serie de trapecios en los que los términos son imprecisos frente a los valores discretos. El sistema permite al experto ajustar los lados del trapecio de la función de pertenencia para que el controlador pueda reaccionar frente a esos valores. Esta funcionalidad permite clasificar toda la base de ejemplos según distintos puntos de vista subjetivos y proceder a obtener resultados en función de lo que, de forma experimental, se persiga.

## 5. Implementación

Un sistema desarrollado en C++ permite efectuar el entrenamiento del controlador, se obtiene la salida nítida  $g_i$  mediante el puerto paralelo de una computadora personal basada en microprocesador Pentium para comprobar los resultados. Como se ha seguido el método basado en las Gradientes Conjugadas, que es un sistema de aprendizaje mediante retroprogramación optimizado y que ofrece una convergencia de la red en un tiempo relativamente aceptable.

La interfase gráfica de usuario se ha desarrollado siguiendo los patrones de Windows de Microsoft para facilitar las pruebas y puesta a punto del controlador.

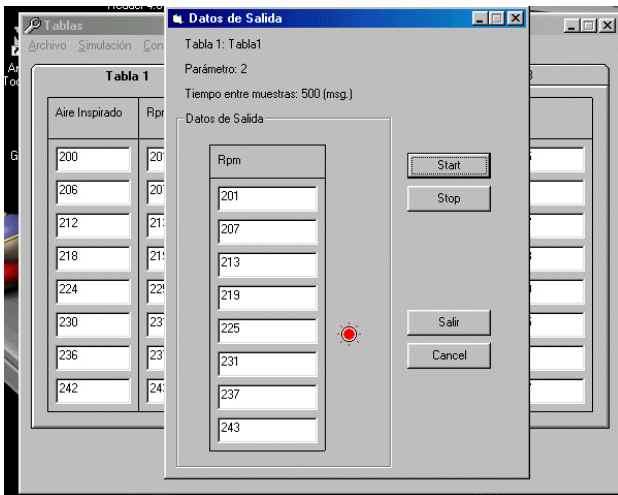


Fig. 1. Ejemplo el interfaz gráfico de usuario donde se permite la inclusión de parámetros de ajuste

El sistema permite introducir, en forma de tablas  $n$ -dimensionales, cualquier tipo de información borrosa a fin de generar unos mapas de superficie sobre los que actuará el controlador hasta enviar la salida al puerto Centronics.

Actualmente, por tanto, el desarrollo está soportado en un ordenador personal con Windows XP para permitir efectuar todo tipo de pruebas y ajustes sin tener que pagar un alto coste en el desarrollo; sin embargo, la versión final tendrá que estar desarrollada dentro de un pequeño equipo, de muy bajo coste, que permita la conversión analógica-digital de varias entradas y viceversa, para su implantación en la industria.

## 6. Conclusiones

El sistema actual, a nivel puramente experimental, aún no ha podido ser comprobado más allá de los casos de prueba definidos con algo más de una decena de datos básicos facilitados. Se está pendiente del desarrollo de una estación de adquisición masiva de información, que debe estar lista para su funcionamiento a primeros de diciembre del presente año, y que facilitará las pruebas de estrés de la configuración. Igualmente se está a la espera de obtener pruebas de rendimiento en el control de un vehículo de inyección electrónica cuyo carburador estará listo en octubre próximo.

Como conclusiones parciales podemos señalar las ventajas proporcionadas en la implantación real del algoritmo de Mandani sobre un procesador Intel estando, como se ha señalado anteriormente, pendientes de poder elaborar pruebas de aprendizaje y de rendimiento del sistema.

## 7. Agradecimientos

El presente artículo ha sido parcialmente subvencionado por el Ministerio de Educación dentro del proyecto de investigación "Desarrollo y verificación de Sistemas Expertos con técnicas lógicas y algebraicas. Teoría y aplicaciones multidisciplinares".

## Referencias

- [1] Klement, E.P.; Mesiar, R: *Triangular norms*. Tatra Mountains. Mathematical Institute Slovak Academy of Sciences, pp. 169-194, Bratislava, 1997.
- [2] Gutiérrez, J.A. et al. *Controlador borroso para la inyección electrónica de motores de dos tiempos*. Internacional Conference Telec'02. Santiago de Cuba, julio de 2002.
- [3] Fahlman S.E.: *Fast-learning Variations on Back-Propagation: an empirical study*. Proceedings of the 1988 Connectionist Models Summer School (Pittsburgh), pp. 38-51. Morgan Kaufmann, San Mateo. USA, 1989.
- [4] Myller M.F.: *A Scaled Conjugate Gradient Algorithm for Fast Supervised Learning*. Research Report, Computer Science Department, University of Aarhus. Alemania, 1990.