

Una Técnica de Inteligencia Artificial para el Ajuste de uno de los Elementos que Definen una B-Spline Racional No Uniforme (NURBS)

Sandra P. Mateus

Politécnico Colombiano JIC, Facultad de Ingenierías, Cra. 48 N° 7-151 P19-146, Medellín-Colombia.
(e-mail: spmateus@elpoli.edu.co).

Recibido Nov. 06, 2009; Aceptado Feb. 08, 2010; Versión Final recibida Abr. 07, 2010

Resumen

Dentro de las técnicas existentes de Inteligencia Artificial, se escogieron y adaptaron dos Redes Neuronales Artificiales (RNA) para realizar el ajuste de uno de los elementos que definen una B-Spline Racional No Uniforme (NURBS) y con ello obtener un modelado adecuado de la NURBS. Los elementos escogidos fueron los puntos de control. Las RNA utilizadas son las de Función de Base Radial y las de Kohonen o Mapas Auto-organizativos. Con base en el análisis de resultados y la caracterización de las RNA, la Función de Base Radial tuvo un desempeño más adecuado y óptimo para un número elevado de datos, lo cual es una desventaja de los Mapas Auto-organizativos. En este modelo se tiene que realizar procesos extras para determinar la neurona ganadora y realizar el reajuste de los pesos.

Palabras Claves: red neuronal artificial, NURBS, puntos de control, funciones de base radial, mapas auto-organizativos.

A Technique of Artificial Intelligence to Fit one of the Elements that Define a Non-Uniform Rational B-Spline (NURBS)

Abstract

In the existing techniques of Artificial Intelligence, two Artificial Neural Networks (ANN) were selected and adapted to fit one of the elements that define a Non-Uniform Rational B-Spline (NURBS) and thus obtaining an appropriate modeling of the NURBS. The selected elements were the checkpoints. The ANN used were the Radial Basis Function and the Kohonen model or Self-Organizing Maps. Based on the analysis of the results and characterization of the ANN the Radial Basis Function had a more appropriate and optimum performance for a large number of data, which is a disadvantage of the Self-Organizing Maps. In this model, additional processes must be done to determine the winning neuron and the weights must be refitted.

Keywords: artificial neural network, NURBS, control points, radial basis functions, self-organizing maps.

INTRODUCCIÓN

Las NURBS (B-Spline Racional No Uniforme) son superficies base empleadas para diversas tareas como la reconstrucción de imágenes 3D y el Diseño Asistido por Computador (CAD/CAM), entre otras. La geometría de las NURBS está determinada por parámetros como los puntos de control y los pesos; dependiendo de la parametrización de éstos, puede existir un número infinito de superficies que ajusten dichos conjuntos. Sin embargo, el gran problema del modelado mediante NURBS está limitado, debido a que la parametrización de los datos que determinan la proximidad de una serie de puntos, es el principal inconveniente para modelar objetos con precisión (Mateus, 2006). Las RNA surgen como una posible solución para resolver problemas de parametrización y presentan características especiales que ayudan a la obtención de superficies a partir de puntos obtenidos.

Se han realizado importantes trabajos sobre la forma de los parámetros adecuados para obtener superficies base a partir de puntos 3D dispersos. En este sentido, Eck y Hoppe (1996) presentaron una solución al problema anterior: la reconstrucción automática de una superficie B-Spline arbitraria; estos autores construyeron una superficie aproximada en forma de malla de un conjunto de puntos desorganizados y posteriormente los proyectaron sobre la malla para obtener sus parámetros, sin embargo, este método requiere que el usuario lo supervise, ya sea por el etiquetado de límites o por el dibujo de frontera en una aproximación de las curvas de la superficie.

Encontrar una superficie base que cumpliera todas las condiciones, posteriormente necesitó de algunas medidas heurísticas. Para superar estos problemas, un nuevo método se presentó, donde una red neuronal artificial fue aplicada para asignar valores a los puntos desorganizados (Hoffmann, 1998). En esta línea, Fabri et al (2000) aplican una red RBF a un modelo Voronoi, obteniendo con ello resistencia al ruido y la construcción de una superficie que aunque lisa, aproxima todos los puntos de los datos, utilizando un número menor de centros en la RBF, lo que conduce a una mayor rapidez en el cálculo de los pesos y una rápida evaluación de las funciones; García (2001) presenta un modelo de RNA de Mapas AutoAsociativos (SOM) basada en segmentos con fase de expansión, para obtener los esqueletos de unos caracteres suministrados de una forma totalmente no supervisada; y Mateus (2006) aplica un SOM a unos puntos dispersos 3D para obtener una aproximación de malla triangular, la cual posteriormente se cuadrilateraliza y en ella se trazan parches NURBS. De los anteriores trabajos, surge la necesidad de trazar NURBS a partir de una organización previa con RNA's de los puntos dispersos.

Con base en lo anterior, esta investigación, trata de dar solución a este problema, adaptando RNA's a los puntos de control de una NURBS, con el objetivo de llegar a un modelado geométrico más adecuado y, apoyados en la literatura revisada, se escogieron las SOM y las RBF como las RNA's para realizar la adaptación en la NURBS. Lo anterior, porque cada vez, se hace más necesario construir modelos desde múltiples mediciones de un objeto real, especialmente cuando la precisión geométrica de un modelo CAD no está disponible o es difícilmente obtenible. Esta necesidad se siente particularmente, cuando se trabaja con objetos de forma libre como esculturas, rostros humanos y muchas otras, que no pueden ser representadas por formas analíticas simples.

En la siguiente sección, se explican las NURBS de forma detallada, haciendo énfasis en los puntos de control. Posteriormente se explica cómo se implementan las RBF y los SOM en los puntos de control de una NURBS. Luego se muestran los resultados y la discusión alrededor de éstos; y por último se exponen las conclusiones.

NURBS

Una curva NURBS Q está definida por (1)

$$S(u, v) = \frac{\sum_{i=0}^p \sum_{j=0}^q B_{i,m}(u) B_{j,n}(v) P_{i,j} W_{i,j}}{\sum_{i=0}^p \sum_{j=0}^q B_{i,m}(u) B_{j,n}(v) W_{i,j}} \quad (1)$$

Donde P_i es un punto de control, W_i es el peso que está asociado a este, y $B_{i,n}$ es una función base definida recursivamente por (2):

$$B_{i,0}(t) = \begin{cases} 1 & \text{si } t_i \leq t \leq t_{i+1} \\ 0 & \text{en otro caso} \end{cases}$$

$$\forall K > 0, B_{i,k}(t) = \frac{t-t_i}{t_{i+k}-t_i} B_{i,k-1}(t) + \frac{t_{i+k+1}-t}{t_{i+k+1}-t_{i+1}} B_{i+1,k-1}(t) \quad (2)$$

las funciones base son vectores de nodos; los valores de estos vectores son ascendentes, es decir $B = \{a_0, a_1=a_0+1, a_2=a_1+1, \dots\}$.

Los puntos de control son una lista de puntos de grado + 1 como mínimo y tienen un número asociado denominado peso; cuando este peso es el mismo en todos los puntos de control (normalmente igual a 1) la curva se denomina racional. Además los puntos de control determinan el orden máximo de la superficie.

Puntos de Control de una NURBS

Una superficie NURBS está completamente determinada por sus puntos de control $P_{i,j}$, lo cual implica, que la superficie cambia de manera predecible conforme lo indican éstos; a esto se le conoce como la propiedad de soporte local y permite que el movimiento de un punto de control sólo afecte la superficie localmente. La principal dificultad en aplicaciones de ajuste es encontrar una parametrización adecuada y escoger automáticamente el número de puntos de control y sus posiciones para definir la topología de la superficie (Ver Figura 1). Los factores de peso $W_{i,j}$ de la superficie NURBS juegan un papel muy importante en el proceso de ajuste, dado que son estos los que determinan que tanto influye localmente un punto en la topología de la superficie. Generalmente, los pesos de los puntos de control de una superficie NURBS se asignan de manera homogénea e iguales a 1, reduciéndose ésta a su caso particular conocido como superficie B-Spline, la cual es limitada en la representación de superficies arbitrariamente curvadas y superficies cónicas.

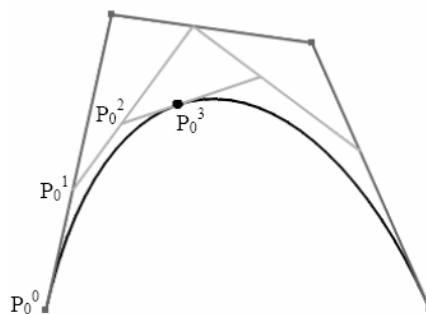


Fig. 1. Polígono de control formado por los puntos de control y la curva NURBS

Intuitivamente, cuanto mayor sea el peso de un punto de control, mayor será la atracción que el punto ejerce sobre la curva. El incremento (o decremento) de un peso produce una aproximación (o un alejamiento) del punto de control asociado. Un peso de valor unidad, no modifica la curva respecto a la no racional. Con pesos mayores de uno, se acerca la curva al punto de control. Con valores menores de uno, la curva se aleja con límite la línea recta que une los puntos de control anterior y siguiente. No se trabaja con pesos negativos. La modificación conjunta de pesos y puntos de control permite modificaciones locales de las curvas y superficies.

Es importante tener en cuenta que al modificar la posición de un punto de control, se produce una modificación local de la superficie. La zona afectada por dicha modificación vendrá dada por el grado de la superficie y de la situación de los otros puntos de control, si se mantienen inalterados los vectores de nodos; es decir, se puede realizar una modificación en un área reducida insertando filas y columnas de puntos de control alrededor de la zona donde se vaya a trabajar, y posteriormente, desplazando los puntos que sean necesarios. Al insertar nuevos puntos de control, la curva no varía;

sin embargo, al eliminar puntos de control se producen variaciones en la curva, pudiendo llegar a ser la curva resultante muy diferente de la curva inicial. Por todo lo anterior, se concluye que los parámetros de las NURBS en el que se va a implementar, la RNA, son los puntos de control.

Existe hoy en día una gran variedad de tipos de redes neuronales, cada una de ellas con una topología y unos métodos de aprendizaje diferentes. Entre las principales arquitecturas empleadas se destacan las redes de Kohonen y las redes de Función de Base Radial. Una de las características más importantes de ambas, es su capacidad de aproximación universal, ya que son capaces de establecer cualquier relación no lineal entre la salida y la entrada con el grado de precisión deseado.

IMPLEMENTACIÓN DE LAS RNA A LOS PUNTOS DE CONTROL DE LA NURBS

Una de las misiones en una red neuronal consiste en simular las propiedades observadas en los sistemas neuronales biológicos a través de modelos matemáticos recreados mediante mecanismos. El objetivo es conseguir que las máquinas den respuestas similares a las que es capaz de dar el cerebro que se caracterizan por su generalización y su robustez (Iglesias, 2001). En el caso de la presente investigación las RNA se relacionan a los pesos que las NURBS necesitan para ajustarse a una geometría dada; dependiendo del aprendizaje de las redes neuronales y los datos de entrada, se dará una salida inteligente. Las RNA están compuestas de un gran número elementos de procesamiento altamente interconectados (neuronas) trabajando al mismo tiempo para la solución de problemas específicos; tal como las personas, éstas aprenden de la experiencia. En cualquier caso, se trata de una forma de computación que es capaz de manejar las imprecisiones que aparecen cuando se trata de resolver problemas relacionados con el mundo real (reconocimiento de formas, toma de decisiones, etc.), ofreciendo soluciones robustas y de fácil implementación. Las RNA han sido entrenadas para la realización de funciones complejas en varios campos de aplicación. Hoy en día pueden ser entrenadas para la solución de problemas que son difíciles para sistemas computacionales comunes o para el ser humano.

Implementación de la red SOM a los puntos de control de la NURBS

Los SOM (Mapas Auto-organizativos o Redes de Kohonen) son una red neuronal no supervisada, competitiva, distribuida de forma regular en una rejilla de, normalmente, dos dimensiones, cuyo fin es descubrir la estructura subyacente de los datos introducidos en ella. A lo largo del entrenamiento de la red, los vectores de datos son introducidos en cada neurona y se comparan con el vector de pesos característico de cada neurona. La neurona que presenta menor diferencia entre su vector de peso y el vector de datos es la BMU (Best Matching Unit - también conocida como neurona ganadora) y ella y sus vecinas verán modificados sus vectores de pesos. Éste tipo de topología, pertenece a la categoría de las redes competitivas o mapas de auto organización, es decir, aprendizaje no supervisado. Poseen una arquitectura de dos capas (entrada-salida) (una sola capa de conexiones), funciones de activación lineales y flujo de información unidireccional (son redes en cascada) (García, 2001).

Las conexiones entre las dos capas de la red dependen de los pesos, en este caso, se iniciaron los pesos con unos valores aleatorios dentro de los rangos $[-1, 1]$ y $[0,1]$. Después de tener la conexión entre las dos capas, los pesos, se pasa a buscar la neurona ganadora; para determinar la neurona ganadora hay dos posibilidades: 1. Determinar la distancia existente entre los pesos y el vector de entrada. 2. Determinar la distancia entre una neurona y su vecina. Para el ajuste de los pesos se tuvo en cuenta dos aspectos:

i) Si la neurona es la neurona ganadora, los pesos de la neurona vecina se ajustan de acuerdo a la ecuación 3:

$$W_{i+1, j+1} = W_{i, j} + a(t) [x(t) - W_{i, j}] \quad (3)$$

donde W son los pesos; a es un valor determinado como la diferencia de 1 menos la iteración sobre el número de datos de entrada.

ii) Si la neurona no es la neurona ganadora, los pesos de la neurona vecina tomarán el valor de dicha neurona (Ver ecuación 4).

$$W_{i+1, j+1} = W_{i, j} \quad (4)$$

El algoritmo base para la implementación del SOM, es el siguiente:

Leer vector de patrones de entrada.

Mientras que # iteraciones < nro. de iteraciones haga

Para a = 0 hasta nodos de entrada, haga.

Para b = 0 hasta nodos de salida, haga

Si # iteraciones j= 0

Si b == neurona ganadora

Pesos_{i+1, j+1} = Pesos_{i, j} + (1 - # iteraciones /nodos de entrada) x Vector de patrones de entrada - Pesos_{i, j}.

Sino

Pesos_{i+1, j+1} = Pesos_{i, j}.

Fin-si

Sino

Pesos_{i, j} = números aleatorios entre [-1, 1].

Fin-si

Fin-para

Fin-para

Determinar la neurona ganadora ().

iteraciones++

Fin-mientras

Determinar la neurona ganadora()

Para a = 0 hasta nodos de salida, haga

Para b = 0 hasta nodos de entrada, haga

Sumatoria += (Pesos_{i, j} - Patrones de entrada)²

Ó Sumatoria += (Pesos_{i, j} - Pesos_{i+1, j+1})²

Fin-para

Distancia_j = $\sqrt{\text{Sumatoria}}$

Fin-para

Auxiliar = 10000

Para a = 0, hasta nodos de salida, haga

Para b = a+1 hasta nodos de salida, haga

Si Distancia [a] < Distancia [b], entonces

Si Distancia [a] < Auxiliar

Auxiliar = a

Fin-si

Sino

Si Distancia [b] < Auxiliar

Auxiliar = b

Fin-si

Fin-si

Fin-para

Fin-para

Implementación de la red RBF a los puntos de control de la NURBS

Las RBF se caracterizan por tener un aprendizaje o entrenamiento híbrido. La arquitectura de estas redes se caracteriza por la presencia de tres capas: una de entrada, una única capa oculta y una capa de salida. En la capa oculta se calcula la distancia entre un vector de pesos y los datos de entrada y sobre esta distancia se realiza una función de tipo radial con forma gaussiana. Para el aprendizaje de la capa oculta hay varios métodos y de este aprendizaje salen unos centros y unas anchuras que darán posteriormente una salida significativa. Posteriormente se pasa a la actualización de los pesos de la capa de salida.

El valor de la distancia d es un componente de la entrada para activar la función radial (Gaussiana). Este es un valor diferencia que tiene las RBF con las demás redes. La función radial (Gaussiana) es una de de las más comunes (Ver ecuación 5):

$$e^{(-r^2)} \quad (5)$$

Siendo r el contenido evaluado en cada nodo de la capa oculta. En este caso particular, el contenido evaluado en cada nodo es la distancia euclidiana d . De ahí la expresión, entonces sería (Ver ecuación 6):

$$e^{(-d^2)} \quad (6)$$

Una de las derivaciones del modelo RBF, es emplear el ancho (desviación estándar) para activar la función radial (Gaussiana). En este caso se trabaja la ecuación 7.

$$e^{\left(\frac{d^2}{a}\right)} \quad (7)$$

donde a es el ancho para ese nodo oculto.

Seguidamente se evaluó cada una de las posibilidades de la función radial (Gaussiana) mostradas anteriormente, para determinar cual presentaba los resultados más adecuados. Dichos resultados, se presentaron, cuando se utilizó como función de activación en la capa oculta, la función radial de la ecuación 4, porque a través de esta función se puede tener la menor distancia hallada entre los centros y los datos alrededor de este y manipular una variable más como el ancho con la cual se pueden manipular los resultados para tener los datos más adecuados para el resultado final.

El algoritmo base para la implementación de la RBF es el siguiente:

Leer vector de centros.

Leer vector de entrada.

Mientras # iteraciones < nro. de iteraciones haga

Si itel= 0, entonces

Para a=0 hasta nodos capa oculta, haga

Para b=0 hasta nodos capa oculta, haga

Sumatoria += (Pesos_{ia,b} - Centros_b)².

Fin-para

Distancia_b = $\sqrt{\text{Sumatoria}}$

Sumatoria = 0

Fin-para

iteraciones++.

Sino

Pesos_{ia,b} = números aleatorios entre [-1, 1].

Fin-si

Fin-Mientras

Para a=0 hasta nodos capa oculta, haga

Neuronas_a = $e^{(-\text{distancia}_a)^2}$

Fin-para

Guardar distancia

RESULTADOS Y DISCUSIÓN

Después de realizar cada una de las pruebas necesarias para el desarrollo de esta investigación, se pasó a la etapa de implementar los valores al parámetro de la NURBS escogido; este caso el parámetro son los puntos de control. El valor en relación con los puntos de control, es la distancia, ya que esta determina la proximidad y el control que tiene un punto sobre otros alrededor de este (Piegl, 2001). A continuación se muestra gráficamente las curvas NURBS obtenidas con los valores de distancia, tanto con la RBF (Ver Figura 2a) como con SOM (Ver Figura 2b), llegando a un resultado

final que concluya que red arroja el dato más adecuado para el parámetro elegido. El número de puntos de control está determinado por el número de neuronas en la capa oculta en el caso de la RBF y el número de neuronas de salida en el SOM. En la Figura 2a, se muestra la curva NURBS obtenida con los datos arrojados por la red RBF para sus puntos de control; en la Figura 2b se muestra la curva NURBS pero con los datos obtenidos con la red SOM.

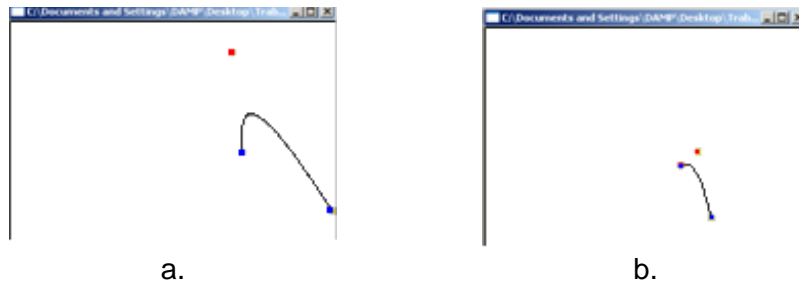


Fig. 2. Curvas NURBS obtenidas con a.RBF b. SOM

Los datos para los puntos de control obtenidos con la RBF son los de la tabla 1 y los datos de los SOM son los de la tabla 2.

Tabla 1. Datos de distancia de la Red RBF

| Nro. | Valor Distancia |
|------|-----------------|
| 1 | 4.22755 |
| 2 | 3.62598 |
| 3 | 9.75165 |

Tabla 2. Datos de distancia de la Red SOM

| Nro. | Valor Distancia. |
|------|------------------|
| 1 | 0.493473 |
| 2 | 2.95398 |
| 3 | 9.18769 |

Las Figuras 2a y 2b son las NURBS con los datos de los puntos de control iniciales; si se realiza una prueba escogiendo el segundo punto de control, que en las figuras 3a y 3b se podrá identificar porque será el que está en color rojo, moviendo el punto de control seleccionado en cada una de las NURBS, se puede observar cual punto de control tiene más importancia en la forma de la NURBS en cada una de las redes neuronales. En la tabla 3 se mostraran los datos iniciales del puntos de control seleccionado y el dato final del mismo punto de control después de realizar un incremento en su valor. El valor a incrementar es igual 1.0.

Tabla 3. Valores iniciales y finales en la red RBF y SOM para el segundo punto de control

| Red RBF | | Red SOM | |
|---------------|-------------|---------------|-------------|
| Valor inicial | Valor Final | Valor inicial | Valor Final |
| 3.62598 | 4.62598 | 2.95398 | 3.95398 |

Los valores de la tabla 3, son los valores iniciales y finales en la red RBF y la red SOM para el segundo punto de control; la diferencia de los dos valores se puede observar entre las figuras 2a y 3a para la red RBF y en entre las figuras 2b y 3b para la red SOM. Si se observa la Figura 2a, la NURBS con el dato inicial del punto de control seleccionado y en la Figura 3a, la NURBS con el dato del punto de control después del movimiento o incremento del valor del punto de control, se puede ver que el movimiento del punto de control realiza un cambio en la curva NURBS, llevando hacia el punto de control el segmento de curva controlado por el punto; sin interferir en los otros puntos de control y sus segmentos de curva.



Fig. 3. Movimiento del Punto de Control con a. RBF b. SOM

En la figura 2b, se observa la NURBS con el dato inicial del punto de control seleccionado y en la figura 3b tenemos la NURBS después del movimiento del punto de control seleccionado; al contrario que en el caso anterior, se puede observar que al modificar el valor del punto de control, no solo se modifica el propio segmento de curva, sino que también se modifica la curva NURBS en general.

CONCLUSIONES

La red neuronal que presenta los resultados más adecuados para implementar en una NURBS, es la RBF, debido a que esta red por sus características y topología, permite manipular los datos de salida esperados en cada una de las capas según las necesidades del caso. La RBF al permitir la posibilidad de calcular la distancia entre tres parámetros diferentes, permite contar con tres posibilidades de valor para la distancia, lo cual aumenta la capacidad de flexibilidad y adaptación a un escenario específico; todo esto en el primer cálculo de la capa oculta. La RBF tiene un desempeño más adecuado y óptimo para un número elevado de datos, lo cual es una desventaja de la red SOM, al tener esta última que realizar procesos extras para determinar la neurona ganadora y realizar el reajuste de los pesos de acuerdo a la neurona ganadora. Como trabajo futuro, se recomienda realizar un estudio donde se pueda implementar una RNA en las NURBS en un espacio tridimensional, que permita el modelado de objetos tridimensionales. Posteriormente del estudio, se podría pensar en el desarrollo de una herramienta que permita el manejo de objetos 3D y si es la necesidad poder corregir distorsiones, por medio de la aproximación de puntos a través de una RNA.

REFERENCIAS

- Eck, M. y H. Hoppe; *Automatic reconstruction of B-Spline surface of arbitrary topological type*. ACM-0-89791-747-4, 8. (1996).
- Fabri, A., y otros autores; *On the Design of CGAL, a Computational Geometry Algorithms Library*. Softw. – Pract. Exp. 30, 11, 1167-1202 (2000).
- García, M.A; *Modelos de Neurocomputación Competitiva para el Descubrimiento de Estructuras*. Tesis Doctoral. Universidad de Málaga (2001).
- Hoffmann M. y L. Várady L; *Free-form Surfaces for Scattered Data by Neural Networks*. Journal for Geometry and Graphics, 2, 1-6. (1998).
- Iglesias, A. y A. Gálvez; *Applying functional networks to fit data points from B-spline surfaces*, in: Magnenat-Thalmann, N. et al. (eds.): Proc. of the Computer Graphics International 2001, Hong-Kong, IEEE CS-Press, 329—332 (2001).
- Mateus, S.; *Método de Generación De NURBS desde Mallas Triangulares para el Ajuste de Superficies de Objetos de Topología Arbitraria*. Tesis de Magister – Universidad Nacional de Colombia, Sede Medellín (2006).
- Piegl, L. y W. Tiller; *Parametrization for surface fitting in reverse engineering*. Computer-Aided Design, 33, 593—603 (2001).