



UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
MAESTRÍA EN MODELOS ALEATORIOS

**Aplicación de un método automático neuro-difuso para la extracción
de carreteras a partir de imágenes satelitales en zonas semi rurales
venezolanas**

*Trabajo de Grado de Maestría presentado ante la ilustre Universidad Central de
Venezuela por el **Lic. Javier Abache Romero**, para optar al Título de
Magister Scientiarium Mención Modelos Aleatorios.*

TUTOR: Dr. José León

Caracas, 21 de marzo de 2012

*Resumen del Trabajo de grado presentado para optar al
título de Magister Scientiarum Mención Modelos
Aleatorios*

**Aplicación de un método automático neuro-difuso para la extracción
de carreteras a partir de imágenes satelitales en zonas semi rurales
venezolanas**

Lic. Javier Abache Romero

Caracas, 21 de marzo de 2012

En este trabajo se presenta un método automático neuro-difuso para la extracción de carreteras a partir de imágenes satelitales pancromáticas; el cual es comparado con un algoritmo difuso. El método consiste de tres pasos: extracción de características, modelado neuro-difuso y extracción de esqueleto. Estos métodos se aplican para extraer las carreteras de una imagen de satélital. Los parámetros de entrada para este sistema son la media y desviación estándar de "ventanas" de 5×5 píxeles de tamaño. En este trabajo, para una mejor detección y la extracción de carreteras, una red difusa se entrena a través de una red neuronal. Los resultados muestran que el algoritmo neuro-difuso proporciona un rendimiento mucho mejor que el algoritmo difuso. Las imágenes utilizadas corresponden a carreteras venezolanas.

Palabras claves: Sistema difuso, sistema neuro-difuso, análisis de imágenes.

Dedicado a mi Madre María Eugenia, a mi Padrino Juan y a mi Tía Magaly.

Agradecimientos

Principalmente agradezco al Profesor José Rafael León por depositar su confianza en mi, aceptando dirigir este trabajo, además, por su esmerada atención y acertadas sugerencias.

A los familiares citados, por su apoyo incondicional.

Índice general

Introducción	VII
1. Lógica difusa	3
1.1. Conjuntos difusos	3
1.1.1. Definiciones básicas	3
1.1.2. Operaciones	5
1.1.3. Funciones de pertenencia	6
1.2. Sistemas de inferencia difusos	8
1.2.1. Principio de extensión	8
1.2.2. Relación difusa binaria	9
1.2.3. Composición de relaciones	9
1.2.4. Variable lingüística	10

1.2.5.	Reglas SI-ENTONCES difusas	10
1.2.6.	Razonamiento difuso	11
1.2.7.	Algunos modelos de inferencia difusos (Mamdani y Sugeno)	13
2.	Redes neuronales artificiales	17
2.1.	Estimador de mínimos cuadrados	18
2.2.	Optimización basada en derivadas	21
2.2.1.	Métodos de descenso	21
2.3.	Redes neuronales adaptativas	22
2.3.1.	Retropropagación en redes de alimentación directa	24
2.3.2.	Regla de aprendizaje híbrida: descenso más rápido y EMC	27
2.4.	Sistemas neuro-difusos de inferencia adaptativa	28
2.4.1.	Algoritmo de aprendizaje híbrido	30
3.	Morfología matemática	33
3.1.	Conceptos básicos de la teoría de conjuntos	34
3.2.	Operaciones morfológicas básicas	34
3.3.	Algoritmo morfológico: Extracción del esqueleto	36

4. Proceso de extracción	37
4.1. Descripción del proceso	38
4.2. Aplicación del proceso. Ejemplos y resultados	41
Consideraciones finales	47
Bibliografía	49

Índice de figuras

4.1. Rural y tierra. Calabozo - Edo. Guárico.	41
4.2. Rural y tierra. Guanare - Edo. Portuguesa	42
4.3. Sub urbano y asfalto. Guacara - Edo. Carabobo	43
4.4. Urbano y asfalto. Riyadh - Arabia Saudita I	44
4.5. Urbano y asfalto. Riyadh - Arabia Saudita II	44
4.6. Sub urbano y asfalto. Las Vegas - E.U. I	45
4.7. Sub urbano y asfalto. Las Vegas - E.U. II	45
4.8. Error de entrenamiento	46

Introducción

Extracción de objetos cartográficos a partir de imágenes digitales — carreteras en nuestro caso — es una operación fundamental para la actualización de un sistema de información geográfica (SIG), sin embargo, no es una tarea sencilla debido a la complejidad de la información contenida en dichas imágenes, las cuales en el campo de la fotometría pueden tener escalas que varían de 1:3000 a 1:90000, mientras que en teledetección la representación en píxeles varían usualmente entre 1 a 30 m (Sowmya, A., Trinder, J., 2000). La estructura de la superficie del terreno en las imágenes comprende una combinación de variadas intensidades que pueden representar áreas verdes, objetos tales como edificios, carreteras, sombras y otros cambios de intensidad de luz; además, cuando se incrementa la escala, la complejidad aumenta aun más debido a la aparición de nuevas características y detalles, lo cual significa que la extracción de objetos en imágenes satelitales enfrenta nuevos retos.

Numerosos métodos para la extracción de estos detalles, se han desarrollado; por ejemplo, basados en segmentación y clasificación: distancias no euclidianas basadas en probabilidad, clasificadores bayesianos, recocido simulado; y basados en lógica difusa (LD) y redes neuronales artificiales (RN), pero de manera general se clasifican en automáticos y semiautomáticos (Mena 2003).

En aquellos, basados en segmentación y clasificación, las variables que caracterizan a las carreteras son numéricas. Imágenes con especificaciones incompletas o ruido son generalmente ignoradas por el método. Por estas razones, se utilizan variables lingüísticas (método basado en LD) para caracterizarlas y una red neural, para controlar las funciones de pertenencia de detección automática de carreteras a partir de imágenes satelitales a escala de grises en alta resolución.

Capítulo 1

Lógica difusa

1.1. Conjuntos difusos

1.1.1. Definiciones básicas

Sea X una colección de objetos y $x \in X$, entonces un **conjunto difuso** A en X es definido como un conjunto de pares ordenados:

$$A = \{(x, \mu_A(x)) \mid x \in X\}, \quad (1.1)$$

donde, $\mu_A : X \rightarrow [0, 1]$ es la **función de pertenencia** de A , que asocia cada punto en X con un número real en el intervalo $[0, 1]$, es decir, μ_A mide el grado de pertenencia de cada elemento de x en A .

La definición de un conjunto difuso, es una extensión de un conjunto clásico, en el cual la función característica tiene valores entre 0 y 1. Generalmente, X se conoce como **universo del discurso**, y puede ser continuo o discreto (ordenado o no),

por lo que A puede escribirse como:

$$A = \begin{cases} \sum_{x_i \in X} \mu_A(x_i)/x_i, & \text{Caso discreto.} \\ \int_X \mu_A(x)/x, & \text{Caso continuo (Ej. Recta real).} \end{cases}$$

Los signos de sumatoria e integral significan la union de los pares $(x_i, \mu_A(x_i))$, al igual que “/” funciona como separador y no implica división.

Un conjunto difuso es identificado de manera única por su función de pertenencia, la cual implica una serie de definiciones explicadas a continuación:

1. **Soporte:** El conjunto soporte de un conjunto difuso A , es aquel que contiene todos los puntos x en X tal que $\mu_A(x) > 0$:

$$\text{soporte}(A) = \{x | \mu_A(x) > 0\}. \quad (1.2)$$

2. **Centro:** Es el conjunto de todos los puntos en X tal que $\mu_A(x) = 1$:

$$\text{centro}(A) = \{x | \mu_A(x) = 1\}. \quad (1.3)$$

3. **Normalidad:** Un conjunto difuso es normal si su centro es no vacío, es decir, siempre podemos encontrar un punto $x \in X$ tal que $\mu_A(x) = 1$.

4. **Conjunto nivel α :** El conjunto *nivel α* de un conjunto difuso A es:

$$A_\alpha = \{x | \mu_A(x) \geq \alpha\}. \quad (1.4)$$

Un conjunto *nivel α* en **sentido fuerte** se define de manera similar:

$$A'_\alpha = \{x | \mu_A(x) > \alpha\}. \quad (1.5)$$

Es claro que, $\text{soporte}(A) = A'_0$ y $\text{centro}(A) = A_1$.

5. **Convexidad:** Un conjunto difuso A es convexo si y solo si para cualquier $x_1, x_2 \in X$ y cualquier $\lambda \in [0, 1]$:

$$\mu_A(\lambda x_1 + (1 - \lambda)x_2) \geq \min\{\mu_A(x_1), \mu_A(x_2)\}. \quad (1.6)$$

Alternativamente, un conjunto es convexo si sus conjuntos *nivel* $-\alpha$ son convexos.

6. **Números difusos:** Un número difuso A en un conjunto difuso en la recta real \Re normal y convexo.
7. **Simetría:** Un conjunto difuso A es simétrico si su función de pertenencia es simétrica alrededor de cierto punto $x = c$, es decir:

$$\mu_A(c + x) = \mu_A(c - x), \forall x \in X. \quad (1.7)$$

8. **Abierto por la izquierda, derecha y cerrado:** Un conjunto difuso A es abierto por la izquierda si $\lim_{x \rightarrow -\infty} \mu_A(x) = 1$ y $\lim_{x \rightarrow \infty} \mu_A(x) = 0$, abierto por la derecha si $\lim_{x \rightarrow -\infty} \mu_A(x) = 0$ y $\lim_{x \rightarrow \infty} \mu_A(x) = 1$; y cerrado si $\lim_{x \rightarrow -\infty} \mu_A(x) = \lim_{x \rightarrow \infty} \mu_A(x) = 0$.

1.1.2. Operaciones

1. **Subconjunto:** Un conjunto difuso A es subconjunto de un conjunto difuso B si y solo si $\mu_A \leq \mu_B$:

$$A \subseteq B \Leftrightarrow \mu_A \leq \mu_B. \quad (1.8)$$

2. **Unión:** La unión de dos conjunto A y B es un conjunto difuso $C = A \cup B$, cuya función de pertenencia es:

$$\mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x). \quad (1.9)$$

Intuitivamente, la union es el conjunto difuso mas pequeño que contiene a A y B .

3. **Intersección:** La intersección de dos conjunto A y B es un conjunto difuso $C = A \cap B$, cuya función de pertenencia es:

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x). \quad (1.10)$$

La intersección es equivalente al conjunto difuso más grande contenido tanto en A como en B .

4. **Complemento:** El complemento de un conjunto difuso A , denotado por $\neg A$ es :

$$\mu_{\neg A}(x) = 1 - \mu_A(x). \quad (1.11)$$

1.1.3. Funciones de pertenencia

Un conjunto difuso es caracterizado completamente por sus funciones de pertenencia. La mayoría de los conjuntos difusos tienen asociado un universo del discurso X , el cual consiste generalmente de la recta real \mathfrak{R} . Es inconveniente listar todos los pares $(x, \mu_A(x))$ que definen una función de pertenencia. Una manera práctica de definirla es formularla matemáticamente. Existen funciones de pertenencia en una y varias dimensiones. Serán descritas solo las de primer tipo; las cuales tienen una sola entrada.

1. **Triangular:** Una función de pertenencia triangular se especifica por tres

parámetros $\{a, b, c\}$ de la siguiente manera:

$$tri(x; a, b, c) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ \frac{c-x}{c-b}, & b \leq x \leq c. \\ 0, & c \leq x. \end{cases}$$

Los parámetros $\{a, b, c\}$ (con $a \leq b \leq c$) son valores en X y determinan los valores de los tres vértices de la una función triangular.

2. **Trapezoidal:** Una función de pertenencia trapezoidal se especifica por cuatro parámetros $\{a, b, c, d\}$:

$$trap(x; a, b, c, d) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ 1, & b \leq x \leq c. \\ \frac{d-x}{d-c}, & c \leq x \leq d. \\ 0, & d \leq x. \end{cases}$$

Los parámetros $\{a, b, c, d\}$ (con $a \leq b \leq c \leq d$) son valores en X y determinan los valores de los cuatro vértices de la una función trapezoidal.

3. **Gaussiana:** Una función de pertenencia gaussiana se especifica por dos parámetros $\{c, \sigma\}$:

$$gaussiana(c, \sigma) = \exp\left(-\frac{1}{2} \left(\frac{x-c}{\sigma}\right)^2\right). \quad (1.12)$$

Donde c determina el centro y σ la amplitud.

4. **Campana generalizada:** Una función campana generalizada se especifica por tres parámetros $\{a, b, c\}$:

$$campana_{gen}(x; a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}}. \quad (1.13)$$

donde, el parámetro b es generalmente positivo. Si b es negativo, la función resulta en una campana que abre hacia abajo.

Esta función, es una generalización de la función de distribución de Cauchy, por lo que también se conoce como función de pertenencia de Cauchy.

1.2. Sistemas de inferencia difusos

1.2.1. Principio de extensión

El principio de extensión provee un procedimiento general para extender dominios determinísticos de expresiones matemáticas a dominios difusos, generalizando una función biunívoca $f(\cdot)$ entre conjuntos clásicos a un mapeo entre conjuntos difusos. Específicamente, supóngase $f : X \rightarrow Y$ y A un conjunto difuso en X definido como:

$$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \cdots + \mu_A(x_n)/x_n. \quad (1.14)$$

Entonces, el principio de extensión establece que la imagen de A bajo $f(\cdot)$ es un conjunto difuso B en Y :

$$B = f(A) = \mu_A(x_1)/y_1 + \mu_A(x_2)/y_2 + \cdots + \mu_A(x_n)/y_n, y_i \in Y, \quad (1.15)$$

donde $y_i \in Y$ es igual a $f(x_i)$, $i = 1, \dots, n$. Es decir, B es definido por medio de $f(\cdot)$ evaluada en x_i . Si $f(\cdot)$ no es biunívoca, existen $x_i \neq x_j \in X$, tales que, $f(x_i) = f(x_j) = y^*$, $y^* \in Y$, donde, $y^* = \mu_A(x_i) \vee \mu_A(x_j)$.

Ahora, supóngase $f : X_1 \times X_2 \times \cdots \times X_n \rightarrow Y$, tal que $f(x_1, \dots, x_n) = y$. Como cada elemento en un vector (x_1, \dots, x_n) ocurre simultáneamente, implica una operación de conjunción. Por lo tanto, el grado de pertenencia de un conjunto difuso B inducido por f es el mínimo de los grados de pertenencia entre los A_i .

1.2.2. Relación difusa binaria

Dados dos universos del discurso X e Y , una relación difusa binaria, es un conjunto difuso en $X \times Y$:

$$R = \{((x, y), \mu_R(x, y)) \mid (x, y) \in X \times Y\} \quad (1.16)$$

Donde, $\mu_R : X \times Y \rightarrow [0, 1]$ y R puede representarse mediante una matriz $|X| \times |Y|$; donde $|\cdot|$ representa la cardinalidad del conjunto \cdot ; siendo r_{ij} , el grado de pertenencia $\mu_R(x_i, y_j)$ del i -ésimo elemento de X y el j -ésimo elemento de Y . Es decir, μ_R es una función de pertenencia bidimensional.

1.2.3. Composición de relaciones

Relaciones difusas sobre productos de universos del discurso pueden ser combinados mediante un operador de composición. Diferentes operadores han sido propuestos y entre los más utilizados, se encuentran la **Composición max-min** y el **Producto-max**.

Sean R_1 y R_2 dos relaciones difusas binarias definidas, en $X \times Y$ y $Y \times Z$:

1. Composición max-min:

$$R_1 \circ R_2 = \{[(x, z), \max_y \min(\mu_{R_1}(x, y), \mu_{R_2}(y, z))] \mid x \in X, y \in Y, z \in Z\}. \quad (1.17)$$

O, equivalentemente:

$$(\mu_{R_1} \circ \mu_{R_2})(x, z) = \bigvee_y [\mu_{R_1}(x, y) \wedge \mu_{R_2}(y, z)], \quad (1.18)$$

donde, \bigvee representa el máximo y \wedge el mínimo.

Cuando R_1 y R_2 se expresan mediante matrices de relación, $R_1 \circ R_2$ equivale

al producto de las matrices que representan cada relación, reemplazando respectivamente, los productos y sumas por \wedge y \vee .

Las relaciones difusas y la composición entre ellas, tienen en común varias propiedades:

Asociatividad: $R \circ (S \circ T) = (R \circ S) \circ T$

Distributividad sobre la unión: $R \circ (S \cup T) = (R \circ S) \cup (R \circ T)$

Distributividad sobre la intersección: $R \circ (S \cap T) \subseteq (R \circ S) \cap (R \circ T)$

Monotonía: $S \subseteq T \Rightarrow R \circ S \subseteq R \circ T$.

2. Composición producto-max: Se define como:

$$(\mu_{R_1} \circ \mu_{R_2})(x, z) = \max [\mu_{R_1}(x, y) \mu_{R_2}(y, z)], \quad (1.19)$$

1.2.4. Variable lingüística

Una variable lingüística se caracteriza por una quintupla $(x, T(x), X, G, M)$, donde, x es el nombre de la variable, digamos *edad*; $T(x)$, es el conjunto término o conjunto de valores lingüísticos: *joven, no tan joven, edad media, no tan viejo, viejo, muy viejo*, etc. G es una regla **sintáctica** que genera los términos en $T(x)$. M es una regla **semántica** que asocia cada valor lingüístico t con su significado $M(t)$, el cual denota un conjunto difuso en X .

1.2.5. Reglas SI-ENTONCES difusas

Una regla SI-ENTONCES difusa (también conocida por regla difusa o implicación difusa), tiene la forma, SI x es A , ENTONCES, y es B o de manera abreviada $A \rightarrow B$. Donde, A y B son valores lingüísticos, definidos, respectivamente, por

conjuntos difusos en los universos del discurso X e Y . “ x es A ” es el antecedente o premisa, e “ y es B ” es el consecuente o conclusión.

Las reglas difusas de la forma $A \rightarrow B$, describen una relación entre dos variables x e y , bajo la forma de una relación difusa binaria R :

$$R = A \rightarrow B = A \times B = \int_{X \times Y} \mu_A(x) \tilde{*} \mu_B(y) / (x, y), \quad (1.20)$$

donde, $\tilde{*} = \min(a, b) = a \wedge b$.

1.2.6. Razonamiento difuso

En lógica tradicional, la regla de inferencia básica, es el **modus ponens**, con la cual podemos inferir la validez de una proposición B de la de A y de $A \rightarrow B$. Sin embargo, en el proceso cotidiano de razonamiento, utilizamos el modus ponens de una manera “aproximada”:

Sean A un conjunto difuso y A' un conjunto difuso “cercano” a A , ambos pertenecientes al universo del discurso X , sean B y B' cercano a B , conjuntos difusos en el universo del discurso Y . Sea $A \rightarrow B$ una implicación expresada como una relación R en $X \times Y$. Entonces el conjunto difuso B' inducido por “ x es A' ” y la regla difusa “SI x es A , ENTONCES, y es B ” es definido por:

$$\mu_{B'}(y) = \max_x \min[\mu_{A'}(x), \mu_R(x, y)] = \vee_x [\mu_{A'}(x) \wedge \mu_R(x, y)], \quad (1.21)$$

o, equivalentemente:

$$B' = A' \circ R = A' \circ (A \rightarrow B). \quad (1.22)$$

A continuación se explicará las reglas de inferencia más comunes:

1. Una regla con simple antecedente: Esta regla corresponde a (1.21), la cual haciendo uso de (1.20) y asociatividad, se puede simplificar:

$$\mu_{B'}(y) = [\forall_x (\mu_{A'}(x) \wedge \mu_A(x))] \wedge \mu_B(y) = w \wedge \mu_B(y), \quad (1.23)$$

donde, w representa el grado de creencia o compatibilidad del antecedente de la regla.

2. Una regla con múltiples antecedentes: Una regla difusa con dos antecedentes tiene la forma “SI x es A y y es B ENTONCES z es C ”. El correspondiente *modus ponens* se expresa:

Premisa 1 (hecho): x es A' y y es B'

Premisa 2 (regla): SI x es A y y es B ENTONCES z es C ,

Consecuencia (conclusión): z es C' .

Premisa 2 (regla), puede reescribirse como “ $A \times B \rightarrow C$ ”. Luego, por (1.20):

$$R_m(A, B, C) = (A \times B) \times C = \int_{X \times Y \times Z} \mu_A(x) \wedge \mu_B(y) \wedge \mu_C(z) / (x, y, z) \quad (1.24)$$

Usando (1.22):

$$C' = (A' \times B') \circ (A \times B \rightarrow C) \quad (1.25)$$

Finalmente, por (1.23):

$$\mu_{C'} = (w_1 \wedge w_2) \wedge \mu_C(z), \quad (1.26)$$

donde, w_1 y w_2 son los grados de compatibilidad o realización; que miden el grado en que la parte antecedente de la regla (Premisa 2) es satisfecha.

3. Varias reglas con múltiples antecedentes: El procesamiento de varias reglas es interpretado, como la unión de las relaciones difusas correspondientes a las reglas difusas. En este caso, el *modus ponens* se expresa:

Premisa 1 (hecho): x es A' y y es B'

Premisa 2 (regla 1): SI x es A_1 y y es B_1 ENTONCES z es C_1 ,

Premisa 3 (regla 2): SI x es A_2 y y es B_2 ENTONCES z es C_2 ,

Consecuencia (conclusión): z es C' .

Usando el modelo descrito en el ítem anterior, podemos escribir la regla 1 como $R_1 = A_1 \times B_1 \rightarrow C_1$ y la regla 2 como: $R_2 = A_2 \times B_2 \rightarrow C_2$. Luego, por (1.25) y la distributividad del operador \circ con respecto al operador \cup :

$$C' = (A' \times B') \circ (R_1 \cup R_2) = [(A' \times B') \circ R_1] \cup [(A' \times B') \circ R_2] = C'_1 \cup C'_2, \quad (1.27)$$

C'_1 , C'_2 y sus respectivas funciones de pertenencia, se calculan con (1.25) y (1.26) respectivamente.

1.2.7. Algunos modelos de inferencia difusos (Mamdani y Sugeno)

Un sistema de inferencia difuso es un *framework* basado en conceptos explicados en la secciones anteriores: conjuntos difusos, reglas SI-ENTONCES y razonamiento difuso.

Tales sistemas, han sido utilizados satisfactoriamente en variedad de áreas; tales como: clasificación, sistemas expertos, predicción en series temporales, robótica, reconocimiento de patrones, etc.

La estructura básica de un sistema de inferencia de este tipo, consiste de tres componentes: Un conjunto de reglas (difusas), una base de datos o diccionario, que define las funciones de pertenencia usadas en las reglas difusas y un mecanismo de inferencia (razonamiento difuso), el cual dado un hecho, en conjunto con las reglas, darán una conclusión. Estos sistemas de inferencia, pueden tomar valores difusos

o numéricos y la salida es generalmente otro valor difuso (en un conjunto); sin embargo, en la práctica, se necesita un valor numérico. Por lo tanto, es necesario, usar lo que se conoce como mecanismo de desfusificación. Existen varios tipos, pero en este trabajo se utilizará el centro de área o centroide. A continuación se explicarán los modelos de inferencia Mamdani, Sugeno y el mecanismo de desfusificación centroide.

- Mamdani: Fue uno de los primeros sistemas de inferencia utilizados. Fue desarrollado por Ebrahim Mamdani en 1975 para controlar un mecanismo de vapor. Su funcionamiento es básicamente lo explicado en el ítem 1.2.6-3. Este sistema arroja solo valores difusos, por lo tanto, es necesario utilizar un mecanismo de desfusificación.
- Desfusificación centroide: Desfusificación se refiere a la forma de como extraer un valor numérico de un conjunto difuso C' , sobre un universo del discurso Z . El mecanismo centroide o centro del área, se lleva a cabo por:

$$z_{COA} = \frac{\int_Z \mu_{C'}(z) z dz}{\int_Z \mu_{C'}(z) dz}, \quad (1.28)$$

donde, $\mu_A(z)$, se obtiene por (1.27). Éste es uno de los sistemas más usados y tiene relación con el cálculo del valor esperado de una función de distribución de probabilidad.

- Sugeno: A diferencia del sistema anterior, el Sugeno no necesita un mecanismo de desfusificación, ya que el valor del consecuente en cada regla es un número.

La forma de una regla en este sistema es “SI x es A y y es B , ENTONCES, $z = f(x, y)$ ”. Donde, f es un polinomio, generalmente de primer grado

(Sugeno de **primer orden**), o de grado cero (**constante** o Sugeno de **orden cero**). La salida final del sistema, es un promedio de los valores de f en cada regla.

Capítulo 2

Redes neuronales artificiales

El problema de determinar un modelo para un sistema desconocido, observando pares de datos entrada-salida (data de entrenamiento) es referido como una identificación de sistema, cuyo objetivo es múltiple:

- Predecir el comportamiento de un sistema
- Estimar las interacciones y relaciones entre entradas y salidas de un sistema.
- Diseñar un controlador basado en el modelo de un sistema.

La identificación de un sistema, comprende dos pasos:

1. Identificación de la estructura: En este paso es necesario aplicar conocimiento *a priori* acerca del sistema que se quiere identificar, con la finalidad de determinar una clase de modelos dentro de la cual se efectuará la búsqueda del más adecuado. Generalmente, esta clase es denotada por una función $y = f(u; \theta)$, donde y es la salida del modelo, u es el vector de entrada

y θ es el vector de parámetros. La determinación de f es dependiente del problema y está basada en parte, en la intuición del que la diseña y las leyes que gobiernan el sistema a identificar.

2. Identificación de parámetros: Una vez conocida la estructura del modelo, es necesario aplicar técnicas de optimización para determinar el vector de parámetros $\theta = \hat{\theta}$, tal que, el modelo resultante $\hat{y} = f(u; \hat{\theta})$, describa el modelo apropiadamente.

2.1. Estimador de mínimos cuadrados

En un problema de mínimos cuadrados, la variable dependiente y de un modelo lineal, es dada por la expresión parametrizada:

$$y = \theta_1 f_1(u) + \theta_2 f_2(u) + \cdots + \theta_n f_n(u), \quad (2.1)$$

donde, $u = [u_1, u_2, \cdots, u_n]^T$ es el vector de entrada del modelo, f_1, f_2, \cdots, f_n son funciones conocidas de u y $\theta_1, \theta_2, \cdots, \theta_n$ los parámetros desconocidos a ser estimados; a la ecuación se le conoce como función de regresión y los θ_i 's coeficientes de regresión.

Para identificar los coeficientes, generalmente se parte de un conjunto de entrenamiento compuesto por pares de datos $\{(u_i; y_i), i = 1, \cdots, m\}$; que representan la relación entrada-salida deseada del sistema a ser modelado. Sustituyendo cada par en (2.1), se genera el siguiente sistema de m ecuaciones lineales:

$$\begin{cases} f_1(u_1)\theta_1 + f_2(u_1)\theta_2 + \cdots + f_n(u_1)\theta_n = y_1, \\ f_1(u_2)\theta_1 + f_2(u_2)\theta_2 + \cdots + f_n(u_2)\theta_n = y_2, \\ \vdots \\ f_1(u_m)\theta_1 + f_2(u_m)\theta_2 + \cdots + f_n(u_m)\theta_n = y_m, \end{cases}$$

matricialmente, el sistema previo, se puede escribir como:

$$A\theta = y, \tag{2.2}$$

con $A_{m \times n}$, llamada, matriz de diseño:

$$A = \begin{bmatrix} f_1(u_1) & \cdots & f_n(u_1) \\ \vdots & \vdots & \vdots \\ f_1(u_m) & \cdots & f_n(u_m) \end{bmatrix}$$

$\theta_{n \times 1}$ es el vector de parámetros desconocidos:

$$\theta = \begin{bmatrix} \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$$

y $y_{m \times 1}$ es el vector de salidas:

$$y = \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix}$$

El conjunto de entrenamiento, compuesto por los pares (u_i, y_i) , es representado por la matriz aumentada $(A|y)$ y el par i de datos de entrada-salida del conjunto por la i -ésima fila $(a_i^T|y_i)$, con $a_i^T = [f_1(u_i), \cdots, f_n(u_i)]$ e $i = 1, \cdots, m$.

Para obtener un conjunto de valores únicos para el vector θ , es necesario que $m \geq n$. Si A es cuadrada ($m = n$) e invertible, podemos resolver (2.2) por:

$$\theta = A^{-1}y, \tag{2.3}$$

Usualmente $m \gg n$, por lo que no es posible obtener una solución exacta que satisfaga las m ecuaciones; ya que la data podría estar contaminada por ruido o el modelo no describir de manera adecuada el sistema a identificar. Es necesario, agregar un término a (2.2) que refleje ese error:

$$A\theta + e = y, \quad (2.4)$$

donde, $e_{m \times 1}$, es el vector de errores y cada componente e_i , representa el error asociado al par entrada-salida $(a_i^T; y_i)$. Ahora, en vez de obtener una solución exacta para (2.2), se busca un vector $\theta = \hat{\theta}$ que minimice:

$$E(\theta) = \sum_{i=1}^n (y_i - a_i^T \theta)^2 = e^T e = (y - A\theta)^T (y - A\theta), \quad (2.5)$$

donde, $e = y - A\theta$ es el vector de errores para una ocurrencia específica de θ .

Si $\hat{\theta}$ es una elección de θ que minimiza (2.5), entonces es llamado, el estimador de mínimos cuadrados (EMC) de θ , el cual satisface:

$$A^T A \hat{\theta} = A^T y. \quad (2.6)$$

Si $A^T A$ es no singular, $\hat{\theta}$ es único y dado por:

$$\hat{\theta} = (A^T A)^{-1} A^T y. \quad (2.7)$$

Geoméricamente, el ECM $\hat{\theta}$, es aquel que garantiza la mejor proyección del vector y , $\hat{y} = A\hat{\theta}$, sobre el espacio generado por la base $A = \{a_1, \dots, a_n\}$; es decir, la proyección, cuyo vector residual e tiene la norma mínima. Dicha proyección, no es más que aquella que garantiza la ortogonalidad de \hat{e} con respecto a \hat{y} .

Desde este punto de vista, el EMC, tiene las siguientes propiedades:

- El error mínimo $E(\hat{\theta})$, dado por (2.5), es equivalente a:

$$E(\hat{\theta}) = y^T \hat{e}. \quad (2.8)$$

- La mejor aproximación de y ; \hat{y} , es ortogonal al residual \hat{e} :

$$y^T \hat{e} = 0. \tag{2.9}$$

2.2. Optimización basada en derivadas

2.2.1. Métodos de descenso

Dada una función real E , definida en un espacio de entrada de n -dimensiones $\theta = [\theta_1, \dots, \theta_n]^T$, se busca un candidato $\theta = \theta^*$ que minimice $E(\theta)$.

Generalmente, E es no lineal y debido a su complejidad, es necesario recurrir a un algoritmo iterativo para explorar el espacio de entrada eficientemente. En métodos de descenso, el siguiente candidato a explorar, θ_{k+1} , para $k = 1, 2, 3, \dots$, es determinado por un “paso” desde el candidato actual, θ_k en un vector director d :

$$\theta_{k+1} = \theta_k + \eta_k d_k, \tag{2.10}$$

donde, k , denota la iteración actual y θ_k y θ_{k+1} , representan dos elementos consecutivos de una secuencia $\{\theta_k\}$ de candidatos a mínimo local θ^* , η representa el tamaño del paso (tasa de aprendizaje en la terminología de redes neuronales). Los métodos de descenso calculan el k -ésimo $\eta_k d_k$, primero hallando el director d y luego calculando η . θ_{k+1} satisface la desigualdad:

$$E(\theta_{k+1}) = E(\theta_k + \eta_k d_k) < E(\theta_k) \tag{2.11}$$

La diferencia entre los diferentes métodos de descenso, viene dada por la forma en que calculan las sucesivas direcciones. Una vez que la decisión es tomada, el siguiente movimiento es determinado por θ_k y el director d , luego el η óptimo se

calcula tomando en cuenta:

$$\eta^* = \arg \min_{\eta > 0} \phi(\eta), \quad (2.12)$$

donde, $\phi(\eta) = E(\theta_k + \eta_k d_k)$.

Por ejemplo, cuando d es determinado en base al vector gradiente g de E , tales métodos de descenso son llamados **métodos basados en el gradiente**.

El gradiente de una función diferenciable $E : R^n \rightarrow R$ en θ , es el vector g de las primeras derivadas de E :

$$g(\theta) = \nabla E(\theta) = \left[\frac{\partial E(\theta)}{\partial \theta_1}, \frac{\partial E(\theta)}{\partial \theta_2}, \dots, \frac{\partial E(\theta)}{\partial \theta_n} \right]^T. \quad (2.13)$$

En general, dado un vector gradiente, las direcciones a elegir dependen de cumplir la siguiente condición:

$$\phi'(0) = \frac{dE(\theta_k + \eta d)}{d\eta} \Big|_{\eta=0} = g^T d = \|g^T\| \|d\| \cos(\xi(\theta_k)) < 0, \quad (2.14)$$

donde, $(\xi(\theta_k))$, denota el ángulo entre g_k y d en θ_k . Si $d = -g$ es la dirección más inclinada en θ_k , es decir, cuando d apunta a $-g$, entonces apunta hacia un mínimo local y en ese caso se habla del **método del descenso más rápido**.

2.3. Redes neuronales adaptativas

Una red adaptativa, es una estructura en red que consiste en un conjunto de nodos conectados por enlaces dirigidos. Cada nodo, representa una unidad de procesamiento y los enlaces, especifican la relación causal entre los nodos. Estos últimos son adaptativos, en el sentido de que su salida depende de parámetros modificables pertenecientes a ellos. Una tasa de aprendizaje η , especifica como esos parámetros debes ser actualizados, para minimizar una medida de error E ,

la cual, es una medida de discrepancia entre la salida k actual de la red y cierta salida deseada. Es decir, una red adaptativa es usada para la identificación de un sistema, descrito por un conjunto de pares de datos de entrada-salida, mediante un conjunto de parámetros (a estimar).

La regla básica de aprendizaje de las redes adaptativas es el método del descenso más rápido, en el cual el vector gradiente es calculado mediante la aplicación sucesiva de la regla de la cadena. Por ejemplo, en las redes multicapas la aplicación de este método, recibe el nombre de retropropagación.

Los parámetros de una red son distribuidos entre los nodos, así que si cada nodo tiene un conjunto no vacío de parámetros se habla de un nodo adaptativo, en caso contrario, se llama nodo fijo.

En general, las redes adaptativas se clasifican de acuerdo a como es la relación entrada-salida de los nodos. Si la salida de cada nodo se propaga sin excepción desde su entrada (izquierda) a su salida (derecha), entonces se trata de una red de *alimentación directa*, por otro lado, si existe un nodo que forma un ciclo, ésta recibe el nombre de recurrente.

Una red de alimentación directa, consiste en un mapeo estático entre conjuntos de entrada y salida, dicho mapeo puede ser lineal o no; va a depender de la estructura de la red y la función asociada a los nodos.

Por ejemplo, la red más simple es aquella con un solo nodo lineal, con dos entradas x_1 y x_2 y una salida x_3 , cuya función es:

$$x_3 = f_3(x_1, x_2; a_1, a_2, a_3) = a_1x_1 + a_2x_2 + a_3,$$

donde, a_1 , a_2 y a_3 son los parámetros modificables y f_3 define un plano, situado arbitrariamente de acuerdo al valor de éstos. Adoptando el error cuadrático como

medida del error y el método de mínimos cuadrados, podemos hallar los valores óptimos de dichos parámetros.

Si agregamos otro nodo a la red, de modo que la salida sean valores en 0, 1, se obtiene:

$$x_4 = f_4(x_3) = \begin{cases} 1, & \text{Si } x_3 \geq 0. \\ 0, & \text{Si } x_3 < 0. \end{cases}$$

Ésta se denomina clasificador lineal; el plano anterior es dividido por una recta, llamada línea de decisión, la cual es determinada por x_3 ; x_4 determina en cual mitad del plano ubicar (x_1, x_2) . Si fusionamos ambos nodos (x_3 y x_4) mediante $f_3 \circ f_4$ obtenemos una red tipo perceptron clásico.

f_4 presenta problemas de discontinuidad, por lo que no es recomendable para aprendizaje basado en derivadas. Por eso debe sustituirse por una aproximación diferenciable, llamada función sigmoide:

$$x_4 = f_4(x_3) \approx \frac{1}{1+e^{-x_3}}$$

2.3.1. Retropropagación en redes de alimentación directa

La regla consiste en como obtener recursivamente, mediante la regla de la cadena el vector gradiente; donde cada una de sus componentes se define como la derivada de una medida de error con respecto a un parámetro. Recibe el nombre de retropropagación, debido a que el gradiente es calculado en la dirección opuesta al flujo de salida de cada nodo. Una vez que el gradiente es calculado, los parámetros son actualizados vía optimización basada en derivadas y regresión.

Supóngase una red adaptativa de alimentación directa con L capas y cada capa $l = 0, 1, \dots, L$; con $l = 0$ representando la capa de entrada; tiene $N(l)$ nodos.

La salida y función del nodo $i = 1, \dots, N(l)$ en la capa l es representado, respectivamente, por $x_{l,i}$ y $f_{l,i}$. Cómo la salida de un nodo depende de sus entradas y parámetros, la función de cada nodo se expresa como:

$$x_{l,i} = f_{l,i}(x_{l-1,1}, \dots, x_{l-1,N(l-1)}, \alpha, \beta, \gamma), \quad (2.15)$$

donde, α, β y γ , son los parámetros del nodo i .

Sea un conjunto de entrenamiento de P entradas. Definimos la medida del error para la p -ésima entrada del conjunto como la suma cuadrática de errores:

$$E_p = \sum_{k=1}^{N(L)} (d_k - x_{L,k})^2, \quad (2.16)$$

donde, d_k es la k -ésima componente del p -ésimo vector de salidas deseadas y $x_{L,k}$ es la salida resultante de suministrar el p -ésimo vector de entrada del conjunto de entrenamiento a la red. Cuando $E_p = 0$, significa que la red es capaz de reproducir la salida deseada del p -ésimo vector del conjunto de entrenamiento. Por lo tanto, la meta es minimizar el error global definido por: $E = \sum_{p=1}^P E_p$. Lo anterior implica que un cambio mínimo en un parámetro α afectará la salida del nodo que lo contiene, a su vez afectará la salida de la última capa y obviamente, la medida del error. Entonces, para calcular el gradiente, es necesario derivar, comenzando en la capa de salida ($l = L$), luego, retrocediendo capa por capa hasta $l = 0$.

Sea $\epsilon_{l,i}$, el error tomado como la derivada de E_p respecto a la salida del nodo i en la capa l :

$$\epsilon_{l,i} = \frac{\partial^+ E_p}{\partial x_{l,i}}, \quad (2.17)$$

donde, ∂^+ significa derivada ordenada. La diferencia entre ésta y la derivada clásica, se encuentra en la forma en como la función es diferenciada. Por ejemplo, para un nodo perteneciente a una capa interna ($l \neq L$), la derivada parcial $\frac{\partial E_p}{\partial x_{l,i}}$ es cero, ya que E_p no depende de $x_{l,i}$ directamente; sin embargo, por lo dicho

anteriormente, este resultado no es cierto, ya que, indirectamente, un cambio en $x_{l,i}$ alterará el valor de E_p .

Por ejemplo, sea $z = g(x, y)$ e $y = f(x)$. En el caso de la derivada clásica $\frac{\partial z}{\partial x}$, la variable y es tomada como constante, sin tener en cuenta que es una función de x . Para la derivada ordenada, esta relación es considerada:

$$\frac{\partial^+ z}{\partial x} = \frac{\partial g(x, f(x))}{\partial x} = \frac{\partial g(x, y)}{\partial x} \Big|_{y=f(x)} + \frac{\partial g(x, y)}{\partial y} \Big|_{y=f(x)} \frac{\partial f(x)}{\partial x}$$

Luego, el error para la salida del i -ésimo nodo de la capa L se calcula directamente:

$$\epsilon_{L,i} = \frac{\partial^+ E_p}{\partial x_{L,i}} = \frac{\partial E_p}{\partial x_{L,i}}, \quad (2.18)$$

para $x_{l,i}$ (nodo interno), el error se calcula, aplicando la regla de la cadena:

$$\epsilon_{l,i} = \underbrace{\frac{\partial^+ E_p}{\partial x_{l,i}}}_{\text{Error capa } l} = \sum_{m=1}^{N(l+1)} \underbrace{\frac{\partial^+ E_p}{\partial x_{l+1,m}}}_{\text{Error capa } l+1} \frac{\partial f_{l+1,m}}{\partial x_{l,i}} = \sum_{m=1}^{N(l+1)} \epsilon_{l+1,m} \frac{\partial f_{l+1,m}}{\partial x_{l,i}}. \quad (2.19)$$

Es decir, el error de un nodo en la capa l , puede ser expresado como una combinación lineal de los errores en $l+1$. Por lo tanto, para cualquier $l = 0, \dots, L$ e $i = 1, \dots, N(L)$, es posible, encontrar $\epsilon_{l,i}$, aplicando (2.18) una vez para encontrar el error en L y (2.19) iterativamente hasta alcanzar l . Este proceso es lo que se conoce como retropropagación, ya que los errores son obtenidos secuencialmente desde $l = L$ hasta $l = 0$.

Para calcular el vector gradiente respecto de un parámetro α de $x_{l,i}$ se utiliza:

$$\frac{\partial^+ E_p}{\partial \alpha} = \frac{\partial^+ E_p}{\partial x_{l,i}} \frac{\partial f_{l,i}}{\partial \alpha} = \epsilon_{l,i} \frac{\partial f_{l,i}}{\partial \alpha}. \quad (2.20)$$

Si α es un parámetro común a varios nodos, (2.20) debe ser modificada a:

$$\frac{\partial^+ E_p}{\partial \alpha} = \sum_{x^* \in S} \frac{\partial^+ E_p}{\partial x^*} \frac{\partial f^*}{\partial \alpha}, \quad (2.21)$$

donde S , es el conjunto de nodos que contienen a α como parámetro; x^* y f^* , son respectivamente la salida y la función de un nodo en S . La derivada del error E global respecto a α es:

$$\frac{\partial^+ E}{\partial \alpha} = \sum_{p=1}^P \frac{\partial^+ E_p}{\partial \alpha}. \quad (2.22)$$

La fórmula de actualización para α resulta en:

$$\Delta \alpha = -\eta \frac{\partial^+ E_p}{\partial \alpha}, \quad (2.23)$$

donde η , es la tasa de aprendizaje y se expresa como:

$$\eta = \frac{\kappa}{\sqrt{\sum_{\alpha} \left(\frac{\partial E}{\partial \alpha}\right)^2}}, \quad (2.24)$$

siendo κ , el tamaño del paso, es decir, la longitud de cada transición a lo largo de la dirección del gradiente en el espacio de parámetros.

2.3.2. Regla de aprendizaje híbrida: descenso más rápido y EMC

Los métodos de descenso más rápido y retropropagación son bastante utilizados en redes adaptativas; sin embargo, toman mucho tiempo en converger. Se puede optimizar el proceso de hallar los parámetros, por ejemplo, teniendo en cuenta que en muchos casos la salida de una red es lineal respecto a varios de esos parámetros; por lo que es posible identificarlos usando el método de mínimos cuadrados.

En consecuencia, dependiendo de los recursos computacionales y el nivel de desempeño requerido, existen al menos cuatro tipos de maneras de combinar estos métodos:

1. EMC solamente: Los parámetros no lineales son fijos mientras que los lineales son identificados por EMC.

2. Descenso más rápido solamente: Todos los parámetros son tratados como no lineales y actualizados por descenso más rápido iterativamente.
3. EMC una vez, seguido de descenso más rápido: El EMC es usado al principio para obtener valores iniciales de los parámetros lineales, luego, descenso más rápido los actualiza iterativamente.
4. Descenso más rápido y EMC: El conjunto de parámetros es discriminado en lineales y no lineales. En cada iteración el descenso más rápido es usado para identificar los no lineales, seguido del EMC para los lineales.

2.4. Sistemas neuro-difusos de inferencia adaptativa

Supóngase un sistema de inferencia difuso *SID* de tipo Sugeno con dos entradas x e y , una salida z y un conjunto de reglas difusas SI-ENTONCES, definidas de la siguiente manera:

Regla 1: SI x es A_1 y y es B_1 , ENTONCES $f_1 = p_1x + q_1y + r_1$,

Regla 2: SI x es A_2 y y es B_2 , ENTONCES $f_2 = p_2x + q_2y + r_2$,

Entonces, un sistema neuro-difuso de inferencia adaptativa (ANFIS por sus siglas en inglés), es una red neural de cinco capas, donde la salida del nodo i y capa l se denota por $O_{l,i}$. Cada una de esas capas es descrita a continuación:

- Capa 1: Cada nodo i en esta capa es adaptativo y su función asociada es:

$$O_{1,i} = \mu_{A_i}(x) \text{ para } i = 1, 2 \dots, \quad (2.25)$$

ó

$$O_{1,i} = \mu_{B_{i-2}}(y) \text{ para } i = 3, 4, \dots, \quad (2.26)$$

donde, x e y son las entradas al nodo i ; A_i ó B_{i-2} son una etiquetas lingüísticas asociadas a ese nodo. Es decir, $O_{1,i}$ es el grado de pertenencia de un conjunto difuso A ($= A_1, A_2, B_1$ ó B_2) y especifica el grado en el cual la entrada x (ó y) satisface lo expresado por A . La función de pertenencia para A puede ser cualquiera de las funciones parametrizadas introducidas en el capítulo 1. Por ejemplo, la función trapezoidal:

$$\text{trap}(x; a, b, c, d) = \begin{cases} 0, & x \leq a. \\ \frac{x-a}{b-a}, & a \leq x \leq b. \\ 1, & b \leq x \leq c. \\ \frac{d-x}{d-c}, & c \leq x \leq d. \\ 0, & d \leq x. \end{cases}$$

donde, $\{a, b, c, d\}$ es el conjunto de parámetros premisa. Por lo tanto, un cambio en ellos, provocará una variación en la forma de pertenencia para A .

- Capa 2: Cada nodo en esta capa es fijo y su salida, es el producto de todas sus entradas:

$$O_{2,i} = w_i = \mu_{A_i}(x) \mu_{B_i}(y), i = 1, 2, \dots. \quad (2.27)$$

Cada salida, representa el grado de compatibilidad o realización (1.26) de una regla.

- Capa 3: Al igual que en la capa anterior, cada nodo es fijo. Ésta vez, la salida es el conciente del grado de compatibilidad de la i -ésima regla entre la sumatoria del grado de compatibilidad de todas las reglas:

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, i = 1, 2. \quad (2.28)$$

- Capa 4: Los nodos de esta capa son adaptativos y su función es definida por:

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i), i = 1, 2, \dots, \quad (2.29)$$

donde \bar{w}_i es (2.28) y $\{p_i, q_i, r_i\}$ es el conjunto de parámetros, llamados consecuente, de este nodo.

- Capa 5: El único nodo de esta capa es fijo y su función calcula la salida global de la red como la sumatoria de todas sus entradas:

$$O_{5,i} = \sum_i \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}. \quad (2.30)$$

De esta manera, se ha construido una red funcionalmente equivalente a un modelo Sugeno. Su estructura no es única, podemos hacer combinaciones de cierta capas para así reducir su número.

2.4.1. Algoritmo de aprendizaje híbrido

De la estructura descrita en la sección anterior, se observa que cuando los valores de los parámetros premisa (Capa 1) son fijos, la salida global es expresada como una combinación lineal de los parámetros consecuentes (Capa 4):

$$\begin{aligned} f &= \frac{w_1}{w_1 + w_2} f_1 + \frac{w_2}{w_1 + w_2} f_2 \\ &= \bar{w}_1 (p_1 x + q_1 y + r_1) + \bar{w}_2 (p_2 x + q_2 y + r_2) \\ &= (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_2, \end{aligned} \quad (2.31)$$

la cual es lineal en los consecuentes p_1, q_1, r_1, p_2, q_2 y r_2 . Ahora, sea S , el conjunto total de parámetros, S_1 conjunto (no lineal) de parámetros premisa y S_2 conjunto (lineal) de parámetros consecuente, es decir, $S = S_1 \cup S_2$. Podemos

entonces, aplicar el algoritmo de aprendizaje híbrido mostrado en la sección (2.3.2). Específicamente, en el paso hacia adelante del algoritmo, las salidas de los nodos van hasta la cuarta capa, entonces los parámetros consecuentes son identificados por mínimos cuadrados. Durante el paso hacia atrás, el error se propaga hasta la primera capa y los parámetros premisa son actualizados vía método del descenso más rápido. Este algoritmo, permite que los parámetros sean identificados de manera óptima ya que se reduce la búsqueda del método de retropropagación clásico. Por lo tanto, es recomendable que S sea descompuesto de la manera mencionada.

Capítulo 3

Morfología matemática

Una imagen digital es el resultado de un proceso de muestreo que consiste en el particionamiento del plano xy en un mallado, cuyo centro es un par de elementos del producto cartesiano Z^2 . Concretamente, una imagen digital es una función $f : Z^2 \rightarrow \mathfrak{R}$, que asigna al par ordenado (Píxel) (x, y) , un valor real que representa una cierta intensidad. La morfología matemática (m.m.) es una poderosa herramienta para el tratamiento de numerosos problemas en el procesamiento de imágenes, bien sea, extracción de componentes o la descripción y representación de ciertas regiones, tales como fronteras, cascós convexos, etc. Su base es la teoría de conjuntos. Estos últimos, representan para la m.m. objetos en la imagen, como por ejemplo, el conjunto de todos los píxeles negros en una imagen binaria, por lo que todos los posibles conjuntos en una imagen, son subconjuntos de Z^2 .

3.1. Conceptos básicos de la teoría de conjuntos

Sea A un conjunto en Z^2 , cuyos elementos son píxeles. Si $w = (x, y)$ es un elemento de A , entonces se expresa como $w \in A$ y $w \notin A$ caso contrario.

Si los elementos de A satisfacen cierta condición, dicho conjunto es representado por comprensión, $A = \{w | \text{condición}\}$. Por ejemplo, el conjunto de elementos que no pertenecen a A , es llamado el complemento de A y denotado por $A^c = \{w | w \notin A\}$.

La unión de dos conjuntos A y B , es un conjunto $C = \{A \cup B\}$ cuyos elementos son aquellos que pertenecen a A , B o a ambos. De manera similar; si $C = A \cap B$ es la intersección de A y B , entonces C contendrá aquellos elementos pertenecientes a A y B simultáneamente.

La diferencia entre dos conjuntos A y B es un conjunto, cuyos elementos son aquellos que pertenecen a A y no a B , es decir, $A - B = \{w | w \in A \text{ y } w \notin B\}$.

Adicional a lo explicado, se definirán dos operadores necesarios para describir posteriormente algunas operaciones morfológicas. Sea $\hat{B} = \{w | w = -b, \text{ con } b \in B\}$.

A este conjunto se le llama reflexión de B . Por otro lado, la traslación de un conjunto A respecto a un punto $z = (z_1, z_2)$, es un conjunto denotado por $(A)_z = \{w | w = a + z, \text{ con } a \in A\}$:

3.2. Operaciones morfológicas básicas

1. Dilatación: Sean A y B conjuntos Z^2 , la dilatación entre ellos se define por:

$$A \oplus B = \left\{ z \mid \left(\hat{B} \right)_z \cap A \neq \emptyset \right\}. \quad (3.1)$$

Es decir, la dilatación consiste en los puntos en común de A y la reflexión de B , seguida de una traslación respecto a z . Su utilidad básica, es la de rellenar agujeros en la imagen.

2. Erosión: Dado dos conjuntos A y B en Z^2 . La erosión es el conjunto:

$$A \ominus B = \{z | (B)_z \subseteq A\}, \quad (3.2)$$

el cual está constituido por todos los puntos z contenidos en A , respecto a los cuales B es trasladado. La erosión es utilizada para eliminar detalles irrelevantes en la imagen.

3. Apertura: La apertura de un conjunto A por un elemento estructurante B es definida por una erosión, seguida de una dilatación:

$$A \circ B = (A \ominus B) \oplus B \quad (3.3)$$

Esta operación, suaviza contornos de un objeto, así como sella fisuras y elimina pequeñas protuberancias.

4. Cierre: El cierre entre A y B , es el contrario a la apertura, es decir, consiste de una dilatación, seguida de una erosión:

$$A \bullet B = (A \oplus B) \ominus B, \quad (3.4)$$

y es utilizada al igual que la anterior, para suavizar contornos, pero a diferencia de la apertura que trabaja por dentro del objeto, ésta trabaja por fuera del mismo.

3.3. Algoritmo morfológico: Extracción del esqueleto

La noción de esqueleto $S(A)$ de un conjunto A es:

- Si z es un elemento de $S(A)$ y $(D)_z$ es el disco de mayor diámetro centrado en z y contenido en A , es imposible encontrar un disco más grande que contenga a $(D)_z$ y esté incluido en A . Por lo tanto, $S(A)$ es la unión mediante líneas de cada uno de los z 's.
- El disco $(D)_z$ toca la frontera de A en dos o más lugares.

El esqueleto de A , puede ser expresado en término de erosiones y aperturas:

$$S(A) = \bigcup_{k=0}^K S_k(A), \quad (3.5)$$

con:

$$S_k(A) = (A \ominus kB) - (A \ominus kB) \circ B, \quad (3.6)$$

donde B es un elemento estructurante y $(A \ominus kB)$ indica k sucesivas erosiones sobre A :

$$(A \ominus kB) = (\dots((A \ominus B) \ominus B) \ominus \dots) \ominus B, \quad (3.7)$$

Y el paso K , es el último de la iteración, antes de que A derive en un conjunto vacío:

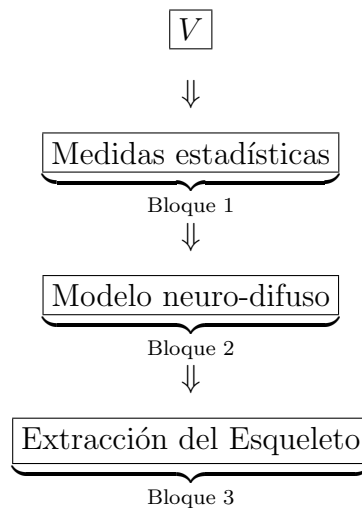
$$K = \max \{k \mid (A \ominus kB) \neq \emptyset\}. \quad (3.8)$$

El conjunto anterior de ecuaciones, establece que $S(A)$ puede ser construido, a partir de la unión de subconjuntos esqueletos $S_k(A)$.

Capítulo 4

Proceso de extracción

Sea $P_{m \times n}$ una matriz que almacena una imagen digital en escala de grises, donde p_{ij} con $i = 1, \dots, m$ y $j = 1, \dots, n$, representa un *pixel*, tomando valores en $[0, 1]$. Luego, P es dividida en un conjunto $V = \{V_k\}$ de "ventanas", cada una de 5×5 *píxeles* de tamaño, con $k = 1, \dots, \frac{1}{25}(m \times n)$. Este conjunto sirve de entrada al proceso de extracción, el cual se puede resumir en el siguiente diagrama:



4.1. Descripción del proceso

- **Bloque 1. Medidas estadísticas:** Entre las características de imágenes, las más utilizadas son aquellas medidas de amplitud, con respecto a la intensidad de luz en algún punto de la imagen (*píxel*) o sobre una vecindad centrada de éste.

Se utilizan en nuestro caso características estadísticas que miden la intensidad y la variación de ésta en cada *píxel*, proporcionando resultados satisfactorios en muchas aplicaciones o procesos de clasificación en imágenes satelitales. Lo cual induce a su uso en extracción y clasificación de objetos espaciales.

La media y la desviación estándar son las características estadísticas utilizadas para este trabajo, y son calculadas para cada uno de los elementos de V .

- **Bloque 2. Modelo neuro-difuso:** Este bloque es subdividido a su vez en tres sub-bloques:

1. Selección del sistema difuso: Una variable numérica (características estadísticas) puede describirse generalmente en términos de un espacio difuso; dividiendo a este último en una serie de conjuntos difusos (1.1) que describen particiones semánticas o estados permitidos de dicha variable numérica.

Por ejemplo; la variable “Media” se divide en tres conjuntos: “pro_car”, “car” y “no_car”. Como la variable almacena el promedio de la intensidad de luz en la imagen, entonces toma valores en $[0, 1]$.

Esta variable, junto a la desviación estándar, constituyen el conjunto de variables lingüísticas de entrada. Adicionalmente, el sistema arroja una variable de salida igualmente lingüística llamada “gs” cuyos conjuntos difusos son: “no_car” y “car”.

Estructuralmente este sub-bloque sigue el concepto de flujo de entrada, proceso, salida: El sistema acepta dos medidas estadísticas conocidas acerca de la imagen, las cuales son convertidas a su representación difusa (*difusificación*). Posteriormente, con esos valores son ejecutadas todas las reglas de inferencia (1.26) y (1.27), resultando luego en un nuevo conjunto difuso para cada variable de salida (Escala de grises). Finalmente, la variable de salida es convertida en un nuevo valor numérico (*desdifusificación*).

2. Evaluación del proceso de extracción: En el sub-bloque anterior se configuró el sistema difuso. En esta parte; una vez que el algoritmo por medio de las funciones de pertenencia (Subsec. 1.1.3), extrae las carreteras, la función de costo - Error cuadrático medio (E.C.M.) - es calculada. Luego, en base al valor de esta última, los resultados son evaluados. Si son satisfactorios, el algoritmo termina. De lo contrario, el proceso pasa al sub-bloque neural.
3. Mejoramiento de las funciones de pertenencia (Sub-bloque neural): Si el resultado de la evaluación no es satisfactorio, se ajustan los parámetros de las funciones de pertenencia (Sec. 2.4). Luego, el algoritmo retorna a los sub-bloques anteriores y los resultados son evaluados nuevamente. El proceso continúa mientras el valor de la función de costo decrece.

- **Bloque 3. Extracción del esqueleto:** Cómo último paso del procedimien-

to, utilizamos morfología matemática (Cap. 3) para la extracción del esqueleto de las carreteras a partir de una imagen binaria generada en el paso anterior.

Morfología matemática es una herramienta para la descripción y representación de regiones en imágenes, tales como, fronteras, esqueletos y cascos convexos.

La teoría subyacente para la morfología matemática es la teoría de conjuntos. Donde “objetos” de interés en una imagen, son representados mediante conjuntos. Por ejemplo; todos los *píxeles* negros en una imagen binaria; donde, cada elemento es un par ordenado (x, y) que representa la coordenada de ese *píxel* en la imagen.

A partir de la imagen resultante del proceso de extracción de carreteras se crea una imagen binaria. De esta última, el esqueleto de las carreteras es extraído, por medio de un algoritmo morfológico.

Un algoritmo morfológico (Sec. 3.3) es aquel que haciendo uso de las operaciones de morfología matemática (Sec 3.2), tales como, erosión, dilatación, apertura y cierre; extrae objetos de interés de una imagen; en la mayoría de los casos, binaria.

Principalmente se considera un algoritmo para la extracción del esqueleto de una región.

4.2. Aplicación del proceso. Ejemplos y resultados

Se aplica el procedimiento descrito a una selección de imágenes que reflejan una combinación de distintos tipos de poblaciones: Urbana, sub urbana y rural;

así como dos tipos de carreteras: tierra y asfalto.

Se agrega para efecto de comparación visual, la imagen resultante de un proceso de extracción de carreteras basado en un modelo difuso simple; es decir, un modelo, donde los parámetros de las funciones de pertenencia y salida no son modificados por una red neuronal.

En cada caso, el orden de presentación es de izquierda a derecha y de arriba a abajo. Comenzando por la imagen a evaluar, siguiendo por el resultado de aplicar el modelo difuso simple, luego el método néuro-difuso y finalmente la extracción del esqueleto.



Figura 4.1: Rural y tierra. Calabozo - Edo.Guárico.

En la figura 4.1 se muestra una zona netamente rural. La carretera principal a detectar es de tierra.

Desde el punto de vista de los valores de intensidad de luz no hay una marcada diferenciación de la carretera con el resto del terreno, observándose un pésimo

desempeño del modelo difuso clásico. Sin embargo, el modelo neuro-difuso la extrajo por completo.

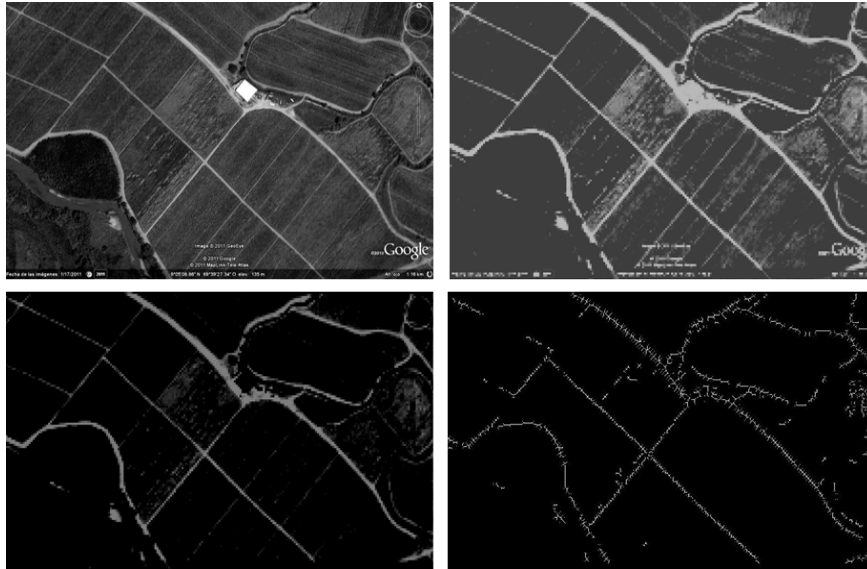


Figura 4.2: Rural y tierra. Guanare - Edo. Portuguesa

La figura 4.2 muestra al igual que en el caso anterior una zona netamente rural con caminos de tierra.

En este caso se observa una marcada diferenciación del camino respecto del resto. El modelo difuso simple muestra un buen desempeño, extrajo la totalidad de los caminos, sin embargo, detectó algunos surcos de arado y otros detalles del terreno, no catalogados como caminos. El modelo neuro-difuso muestra un mejor resultado, ya que si bien extrajo otros elementos, aparecen con menor intensidad, siendo luego ignorados en el proceso de detección del esqueleto.

Las imágenes que siguen a continuación, corresponden a densidades sub-urbanas y urbanas. Aunque están fuera del ámbito de este trabajo, se incluyen para mostrar la fortaleza del método neuro-difuso.

Cabe destacar, que en otras investigaciones, para obtener resultados ligeramente

mejores a los que se presentarán, las imágenes pasan por etapas de pre procesamiento o por intervención humana en algún punto.

En cada una de las imágenes se observa que las carreteras están bien diferenciadas.

El modelo difuso simple extrae los caminos completamente, pero el neuro-difuso elimina detalles que en el anterior se tomaron como caminos.

En todos los casos persisten detalles falsamente detectados, por ejemplo, azoteas de edificios y arbustos, que en una imagen en escala de grises poseen el mismo nivel de intensidad de luz que las carreteras.

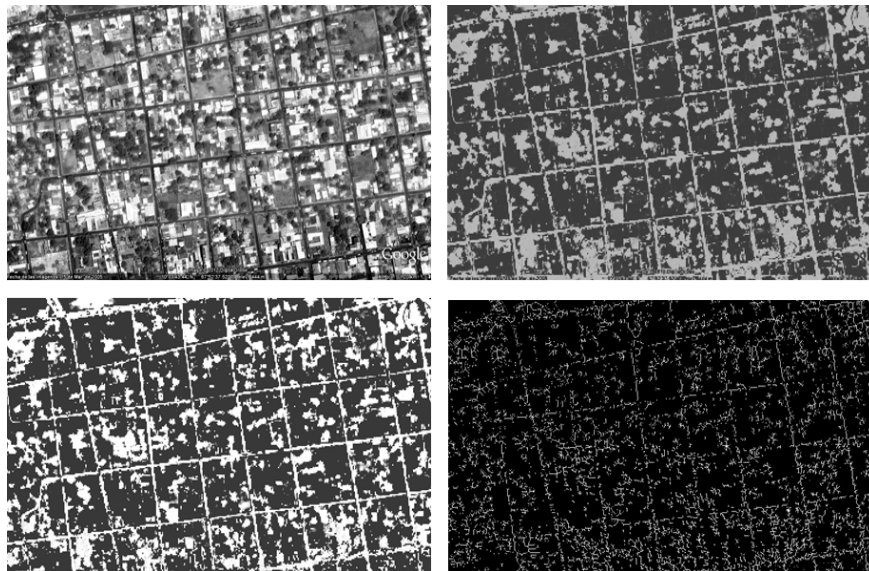


Figura 4.3: Sub urbano y asfalto. Guacara - Edo. Carabobo

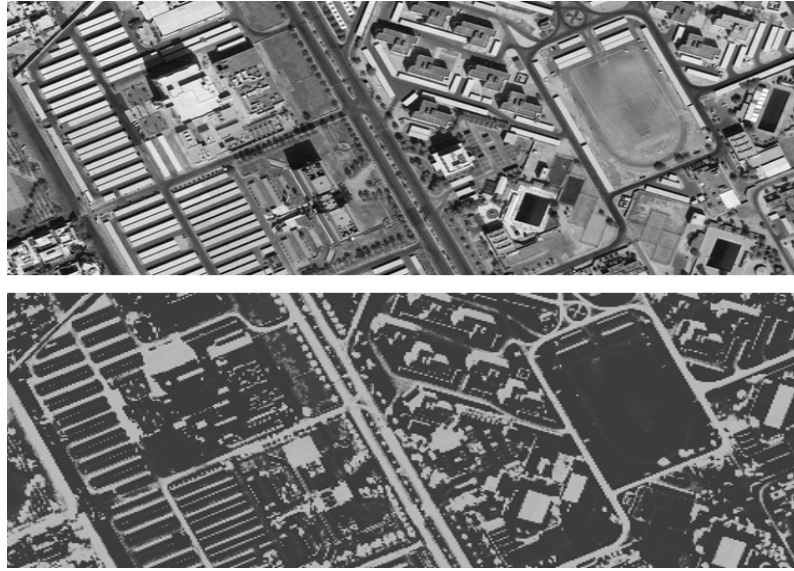


Figura 4.4: Urbano y asfalto. Riyadh - Arabia Saudita I

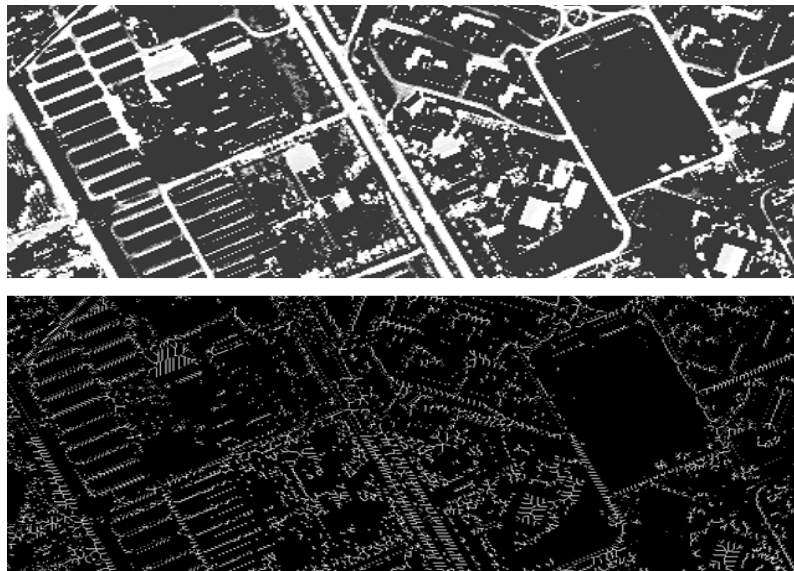


Figura 4.5: Urbano y asfalto. Riyadh - Arabia Saudita II



Figura 4.6: Sub urbano y asfalto. Las Vegas - E.U. I



Figura 4.7: Sub urbano y asfalto. Las Vegas - E.U. II

Finalmente, se presenta la gráfica del error de entrenamiento de la red. El patrón para entrenarla consta de tres columnas: las dos primeras, son respectivamente, el valor de la media y desviación estándar de una ventana y la tercera toma un valor de uno si a partir de los dos valores anteriores, se trata de un camino y cero en caso contrario. Cabe destacar que la data de entrenamiento se dividió en dos conjuntos: uno para entrenar y el otro para verificación de la existencia de sobreajuste. Por lo que en la gráfica se observan dos curvas; la azul, representa el error de entrenamiento y la verde; el error que vigila el sobreajuste, el cual, como se puede constatar no se presenta, ya que la curva es descendente durante todo el entrenamiento.

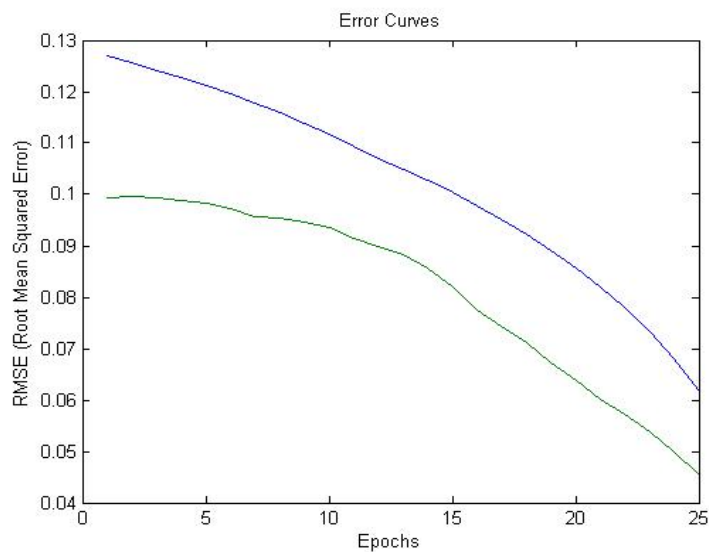


Figura 4.8: Error de entrenamiento

Consideraciones finales

En este trabajo se presentó un método neuro difuso para la extracción de carreteras en imágenes satelitales en escala de grises de alta resolución.

Una imagen sirve de entrada al sistema y es dividida para ello en ventanas de 5×5 píxeles; a cada ventana le es calculada la media y desviación estandar. Estas medidas son transformadas a su representación difusa y los valores en cada ventana son suministrados a un sistema difuso de inferencia.

Una vez que todas las ventanas son procesadas, se evalúa la calidad de los resultados por medio de la función de costo (Error cuadrático medio). Si son satisfactorios, el algoritmo termina, de lo contrario una red neuronal es utilizada para mejorar los parámetros de las funciones de pertenencia y salida.

A la imagen resultante le es extraído el esqueleto mediante el uso de la morfología matemática.

Adicionalmente a la función de costo, el método es comparado con un sistema difuso clásico, es decir, aquel donde los parámetros de las funciones de pertenencia no son sometidos a un proceso de aprendizaje mediante una red neural. Considerando los resultados, nuestro método presenta un mejor desempeño que el difuso simple, debido precisamente al perfeccionamiento de los parámetros del sistema por medio de la red neural.

Bibliografía

- [1] J. Amini (2006), *A fuzzy model for identification in satellite images*, Proceedings of the 2006 International Conference on Image Processing, Computer Vision and Pattern Recognition, Las Vegas, Nevada, USA, June 26-29.
- [2] Z. Chi, H. Yan, T. Phan (1996), *Fuzzy algorithms: with applications to image processing and pattern recognition*, World Scientific.
- [3] R. Fullér (1995), *Neural fuzzy systems*, Abo Akademi University.
- [4] R. Gonzalez, E. Woods (1992), *Digital Image Processing*, Addison-Wesley, Reading, MA.
- [5] J. Horváth (2006), *Image segmentation using fuzzy c-means*, preprint.
- [6] J. Jang, C. Tsai Sun, E. Mizutani (1997), *Neuro Fuzzy and Soft Computing*, Prentice Hall Upper Saddle River, NJ 07458.
- [7] T. Knudsen, H. Muhammed, B. Olsen (2002), *A comparison of neuro fuzzy and traditional image segmentation methods for automated detection of buildings in aerial photos*, ISPRS, Photogrammet, Comput, Vision, 913 September, Graz, Austria, B-116 p.

- [8] J. Mena, J. Malpica (2005), *An Automatic Method for Road Extraction in Rural and Semi-Urban Areas Starting from High Resolution Satellite Imagery*, Pattern Recognition Letters 26, pp. 1201-1220.
- [9] J. Mena (2006), *State of the art on automatic road extraction for GIS update: a novel classification*, Pattern Recognition Letters 24 (2003) 30373058, Elsevier.
- [10] J. Serra (1982), *Image Analysis and Mathematical Morphology*, Academic Press.
- [11] P. Soille (2004), *Morphological Image Analysis: Principles y applications*, Springer.
- [12] L. Zadeh (1965), *Fuzzy sets*, Inform, Control 8, 338-353.
- [13] A. Zare (2010), *Automatic road extraction based on neuro-fuzzy algorithm*, Proceedings of the 10th WSEAS Int. Conference on Robotics, Control and Manufacturing Technology.