

UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE CIENCIAS



TESIS

**“MODELACIÓN MATEMÁTICA NEURO-DIFUSA
PARA SIMULAR EL CAUDAL Y NIVEL DEL FLUJO
DE AGUA EN LOS RÍOS”**

**PARA OBTENER EL TÍTULO PROFESIONAL DE LICENCIADO EN
MATEMÁTICAS**

**ELABORADA POR:
ANGELLO MORANTE MORAN
Asesora: Dra. IRLA MANTILLA NÚÑEZ**

LIMA - PERÚ

2019

Agradecimientos

Mi más sincero agradecimiento a la profesora Dra. Irla Mantilla Núñez por su dedicación, tiempo e invaluable apoyo durante el desarrollo de este trabajo, así como por haberme facilitado el uso de los equipos e instalaciones del Laboratorio de Simulación e Investigación Numérica de la Facultad de Ciencias de la Universidad Nacional de Ingeniería.

A mis padres, por los valores que
me han inculcado y han hecho de mi una persona de bien.

Resumen

Las Redes Neuronales Artificiales, constituyen una herramienta que se ha aplicado en las últimas décadas a distintos campos del conocimiento (desde la Robótica a la Psicología), con buenos resultados. A partir de 1991 aparece unas de las primeras aplicaciones en Hidrología; como predicción de lluvia, consumo de agua, estimación y predicción de caudales.

En esta tesis, se ha incorporado herramientas de inferencia difusa a una red neuronal artificial recurrente, lo que origina el concepto de Red Neuronal Difusa Recurrente. Este tipo de red tiene la ventaja de ser más adaptable que una red neuronal usual para el estudio de sistemas caóticos. En primer lugar, porque escogiendo funciones de pertenencia triangular y trabajando sin capas ocultas, la función de error es cuadrática respecto a los pesos sinápticos; lo cual facilita el hallazgo de un valor mínimo global mediante el método de la pendiente máxima. En segundo lugar, porque para encontrar su funcionamiento óptimo; en lugar de cambiar el número de capas ocultas, como se hace en una red neuronal usual; se puede trabajar sin capas ocultas y cambiar solo el número de funciones de pertenencia involucradas, reduciendo el costo computacional.

Como aplicación de la metodología mencionada en el párrafo anterior; al considerar el caudal y nivel del flujo de agua de un río como variables almacenadas en registros hidrométricos históricos, que funcionan como una serie de tiempo con un orden de retraso finito; se podrá modelar el comportamiento de dichas variables para su monitoreo a corto plazo.

Palabras Clave: Redes neuronales, algoritmo de aprendizaje, inferencia difusa, registro hidrométrico.

Abstract

Artificial Neural Networks constitute a tool that has been applied in the last decades to different fields of knowledge (from Robotics to Psychology), with good results. Since 1991, one of the first applications in Hydrology has appeared; such as rainfall forecasting, water consumption, estimation and prediction of flows.

In this thesis, fuzzy inference tools have been incorporated into a recurrent artificial neural network, which originates the concept of Recurrent Fuzzy Neural Network. This type of network has the advantage of being more adaptable than a usual neural network for the study of chaotic systems. First, because by choosing triangular membership functions and working without hidden layers, the error function is quadratic with respect to the synaptic weights; which makes easier the task of finding a minimum value by the steepest descent method. Secondly, because to find its optimal functioning; instead of changing the number of hidden layers, as is done in a conventional neural network; one can work without hidden layers and change only the number of membership functions involved, reducing the cost computational.

As an application of the methodology mentioned in the previous paragraph; when considering the flow and level of a river's water flow as variables stored in historical hydrometric records as a time series with a finite delay order; the behavior of these variables could be modeled for short-term monitoring.

Keywords: Neural networks, learning algorithm, fuzzy inference, hidrometric record.

Contenido

Resumen	II
Lista de figuras	V
Lista de tablas	VII
Introducción	IX
1. Fundamentos Matemáticos	1
1.1. Medida de Lebesgue en \mathbb{R}^n	1
1.2. Método de la Pendiente Máxima	4
1.3. Redes Neuronales Artificiales	5
1.3.1. Capacidades de aproximación de las redes neuronales	15
1.3.2. Algoritmo de Aprendizaje de una Red Neuronal Artificial	20
1.3.3. Redes Neuronales Recurrentes	23
1.4. Lógica Difusa	26
1.4.1. Conjuntos Difusos	26
1.4.2. Inferencia Difusa	31
1.4.3. Ejemplo de aplicación: Uso de Inferencia difusa para diseñar un sistema de control difuso de una grúa. ([9])	38
2. Fundamentos Hidrológicos Generales	44
2.1. Nociones Básicas de Hidrología	44
2.2. Aspectos Generales de los Procedimientos Hidrométricos.	47
2.2.1. Requerimientos para la selección de la sección de aforo.	49
2.2.2. Demarcación de la sección de aforo para el estudio del problema.	50
2.2.3. Estaciones Hidrométricas	51
2.2.4. Equipos de medición	52

2.3. El río Rímac	56
3. Planteamiento del Problema	58
3.1. Técnicas usadas actualmente por SENAMHI.	59
3.2. La propuesta de trabajo	60
4. Una nueva metodología: Red Neuro-Difusa Recurrente.	62
4.1. Un modelo neuronal difuso	62
4.2. Modelo de Red Neuronal Difusa Recurrente	65
4.2.1. Estructura de una Red Neurodifusa Recurrente	65
4.2.2. Algoritmo de aprendizaje de una Red Neurodifusa Recurrente.	67
5. Resolución Numérica del Problema y Resultados	69
5.1. Obtención de datos y región de estudio	69
5.2. Estructura Neuronal Difusa usada para el modelamiento.	74
5.3. Implementación y Resultados.	74
6. Conclusiones	89
Bibliografía	103

Índice de figuras

1.1. Partes principales de una neurona biológica	5
1.2. Sinapsis	6
1.3. Representación gráfica del perceptrón simple	11
1.4. Representación gráfica del perceptrón multicapa	14
1.5. Red neuronal recurrente.	24
1.6. Descarga de contenedores	38
2.1. Principales Rios del Perú	46
2.2. Vertientes Hidrográficas del Perú	48
2.3. Correntómetro	53
2.4. Contador	53
2.5. Limnímetro	54
2.6. Limnígrafo	55
2.7. Limnigrama	55
2.8. Malacate	56
2.9. Tarabita	57
4.1. Red Neuronal Difusa	63
4.2. Funciones de pertenencia	63
4.3. Red Neuronal Difusa recurrente	66
5.1. Cuenca del rio Rimac	70
5.2. Estación San Mateo	71
5.3. Estación Río Blanco	72
5.4. Estación Chosica	73
5.5. Red Neuronal Difusa Recurrente usada para el modelamiento	74
5.6. Diagrama de flujo	77
5.7. Caudal Estación San Mateo	79

5.8. Error de entrenamiento. Estación San Mateo	80
5.9. Caudal Estación Río Blanco	81
5.10. Error de entrenamiento. Estación Río Blanco	82
5.11. Nivel Estación San Mateo	83
5.12. Error de entrenamiento. Estación San Mateo	84
5.13. Nivel Estación Río Blanco	85
5.14. Error de entrenamiento. Estación Río Blanco	86
5.15. Nivel Estación Chosica	87
5.16. Error de entrenamiento. Estación Chosica	88
6.1. Diagrama de flujo para obtener los pesos óptimos al entrenar la red . . .	102

Índice de cuadros

5.1. Valores máximo y mínimo del caudal (m^3/s) en las Estaciones San Mateo y Río Blanco.	75
5.2. Valores máximo y mínimo del nivel ($dm = 0,1m$) en las Estaciones San Mateo, Río Blanco y Chosica.	75
5.3. Valores de los parámetros usados en el modelamiento del caudal.	75
5.4. Valores de los parámetros usados en el modelamiento del nivel.	76
5.5. Evaluación del modelamiento del caudal por medio de los errores obtenidos (m^3/s)	78
5.6. Evaluación del modelamiento del nivel por medio de los errores obtenidos (dm)	79

Introducción

El uso de las redes neuronales en Hidrología es un acontecimiento reciente, pero ha tenido una rápida aceptación y un florecimiento positivo en publicaciones y conferencias. Por otro lado, la Lógica Difusa es una herramienta matemática que nació hace medio siglo, con la finalidad de dar un tratamiento matemático más preciso a muchas nociones de pertenencia vagamente definidas en la vida real. Como resultado de tal, el tratamiento de información con un comportamiento altamente caótico puede ser realizado de una manera más simple. Así es como nace una nueva herramienta que recibe el nombre de Red Neuro-Difusa y las publicaciones al respecto solo tienen alrededor de 10 años de antigüedad ([13], [16], [21],[22], [23],[24]). En esta tesis se hace uso de esta herramienta joven con la finalidad de hacer predicciones a corto plazo del nivel del flujo de agua y caudal de un río.

En el primer capítulo se desarrollan las herramientas matemáticas necesarias para la elaboración de la metodología a usar en el capítulo 4. Aquí se puede distinguir dos aspectos principales: los fundamentos matemáticos de las Redes Neuronales ([4], [17], [18], [19], [20]) y la Lógica Difusa ([5], [6], [7], [8], [9]). En cuanto a redes neuronales, se explica el uso del método de la pendiente máxima en la elaboración del algoritmo de aprendizaje de una red neuronal, asimismo las deducciones de las ecuaciones de dicho algoritmo. También se explica el algoritmo de aprendizaje de una red neuronal recurrente, es decir, una red neuronal en la que las salidas vuelven a funcionar como entradas de la misma red. En cuanto a lógica difusa, se aborda las definiciones y propiedades básicas de los conjuntos difusos, y se explica el mecanismo de inferencia difusa, el cual en resumen, es un proceso a través del cual un valor de entrada es fusificado, es decir ubicado en un conjunto difuso, y a través de una serie de reglas es transformado en un valor de salida perteneciente a otro conjunto difuso para luego ser desfusificado y obtener un valor numérico concreto. Dicho mecanismo sirve para modelar sistemas en los

que es muy difícil obtener una fórmula explícita de las variables de salida en términos de ciertas variables de entrada.

En el capítulo 2, se aborda lo necesario sobre la terminología usada en hidrología, así como los aspectos principales sobre los procedimientos hidrométricos y su importancia.

En el capítulo 3, se formula el planteamiento del problema explicando las suposiciones básicas sobre las que se va a trabajar. Principalmente se considera que los valores de niveles y caudales pueden predecirse (al menos a corto plazo) por una función continua, cuya formulación puede ser tan complicada que sugiere la intervención de un sistema de inferencia difuso para su formulación. A su vez, dado que los modelos actuales necesitan de otros datos adicionales (lluvia, escorrentía, etc.) y lo que se pretende en esta tesis es formular un modelo sencillo, entonces se considera también que al usar una red neuronal recurrente, dichas dependencias serán incluidas implícitamente en nuestro modelo.

En el capítulo 4 se detalla el funcionamiento de una red neuronal difusa recurrente, cuya formulación fue motivada por las razones explicadas en el párrafo anterior. Además, con la experiencia adquirida en el desarrollo del primer capítulo, se deduce las ecuaciones de su algoritmo de aprendizaje.

En el capítulo 5 se muestra los resultados de la implementación numérica de las ecuaciones obtenidas en el capítulo 4, usando los datos proporcionados por el Servicio Nacional de Meteorología e Hidrología del Perú (SENAMHI).

Finalmente, en el capítulo 6 se enumeran las conclusiones del trabajo y se habla sobre algunas expectativas futuras sobre el tema.

Capítulo 1

Fundamentos Matemáticos

1.1. Medida de Lebesgue en \mathbb{R}^n

Definición 1.1.1. (σ -álgebra y medida). Sea Ω un conjunto no vacío. Una familia Σ de subconjuntos de Ω recibe el nombre de σ -álgebra si verifica las siguientes propiedades:

- i) $\Omega \in \Sigma$.
- ii) $A \in \Sigma \Rightarrow \Omega - A \in \Sigma$.
- iii) $[A_n \in \Sigma, \forall n \in \mathbb{N}] \Rightarrow \bigcup_{n=1}^{\infty} A_n \in \Sigma$.

La pareja (Ω, Σ) se denomina espacio medible y los elementos de Σ se denominan conjuntos medibles. Sea (Ω, Σ) un espacio medible. Una medida sobre Σ es una aplicación $\mu : \Sigma \rightarrow [0, \infty]$ que verifica la siguiente propiedad conocida como σ -aditividad

$$[A_n \in \Sigma \forall n \in \mathbb{N}, A_i \cap A_j = \emptyset \forall i \neq j] \Rightarrow \mu \left(\bigcup_{n=1}^{\infty} A_n \right) = \sum_{n=1}^{\infty} \mu(A_n).$$

Para evitar situaciones triviales, se supone siempre que existe $A \in \Sigma$ tal que $\mu(A) < \infty$. La terna (Ω, Σ, μ) se denomina espacio de medida.

Sea Ω un conjunto no vacío cualquiera. La familia $P(\Omega)$ constituida por todos los subconjuntos de Ω es una σ -álgebra en Ω . Si $\{\Sigma_i : i \in I\}$ es una familia de σ -álgebras en Ω , entonces $\Sigma = \bigcap_{i \in I} \Sigma_i$ es una σ -álgebra en Ω . En consecuencia si $S \subset P(\Omega)$, entonces la intersección de todas las σ -álgebras en Ω que contienen a S (entre las cuales hay al menos una: $P(\Omega)$) es la menor σ -álgebra en Ω que contiene a S y recibe el nombre de σ -álgebra generada por S .

Ejemplo 1.1.1. Sea Ω un conjunto.

1. Si A es un subconjunto no vacío de Ω y $\mathcal{F} = \{\emptyset, \Omega, A, A^c\}$ entonces \mathcal{F} es la σ -álgebra generada por $S = \{A\}$.
2. Si S es vacío, o $S = \{\emptyset\}$ o $S = \{\Omega\}$, entonces la σ -álgebra generada por S es igual a $\{\emptyset, \Omega\}$.
3. Si $S \subset \mathcal{P}(\Omega)$ es la familia de subconjuntos de Ω que constan de un solo elemento, entonces la σ -álgebra generada por S es:

$$\{A \subset \Omega : A \text{ es numerable}\} \cup \{A \subset \Omega : A^c \text{ es numerable}\}$$

LA σ -ÁLGEBRA DE BOREL. La σ -álgebra de Borel en \mathbb{R}^n es la σ -álgebra generada por la familia de todos los conjuntos abiertos de \mathbb{R}^n . En lo sucesivo se denotará por \mathcal{B} y sus elementos se denominarán conjuntos borelianos.

Obsevación 1.1.1. La σ -álgebra de Borel de \mathbb{R}^n también puede ser generada por la familia de conjuntos de la forma $\prod_{i=1}^n \langle -\infty, x_i \rangle$, donde $x_1, x_2, \dots, x_n \in \mathbb{R}$.

Es claro que los conjuntos cerrados son borelianos. De hecho \mathcal{B} contiene las siguientes familias de subconjuntos topológicamente relevantes:

- i) Conjuntos del tipo G_δ que son aquellos conjuntos que se pueden expresar como intersección numerable (y decreciente, si se quiere) de abiertos.
- ii) Conjuntos del tipo F_σ que son aquellos conjuntos que se pueden expresar como unión numerable de cerrados. Es inmediato comprobar que un conjunto del tipo F_σ se puede expresar como unión numerable y creciente de conjuntos compactos.

Definición 1.1.2. (Propiedad c.p.d.). Sea (Ω, Σ, μ) un espacio de medida. Se dice que una propiedad $P(w)$ relativa a un punto genérico $w \in \Omega$ se verifica casi por doquier (lo cual abreviaremos por c.p.d.) con respecto a μ si el conjunto de puntos donde dicha propiedad no se verifica es un conjunto de medida cero. Se omite la referencia a la medida μ si no hay lugar a confusión.

Definición 1.1.3. (Volumen de un intervalo de dimensión n). Un intervalo en \mathbb{R}^n es un conjunto de la forma $I = I_1 \times \dots \times I_n$ donde I_1, \dots, I_n son intervalos (cada uno

de ellos de cualquier tipo, abierto, cerrado, semiabierto, acotado o no) de \mathbb{R} . Cuando el intervalo I es acotado y no vacío se define su volumen por:

$$v(I) = (\sup I_1 - \inf I_1) \cdots (\sup I_n - \inf I_n).$$

Por coherencia convenimos que $v(\emptyset) = 0$. Si $n = 1$ el volumen 1-dimensional de un intervalo acotado $I \subset \mathbb{R}$ se llama la longitud de I y se representa por $l(I)$. Para $n = 2$ ($n = 3$), el volumen 2-dimensional (3-dimensional) de un intervalo acotado en \mathbb{R}^2 (\mathbb{R}^3), que no es otra cosa que un rectángulo (ortopedro) de lados paralelos a los ejes, es el área (volumen) de dicho rectángulo (ortopedro).

Definición 1.1.4. (Medida exterior de Lebesgue en \mathbb{R}^n). La medida exterior de Lebesgue, ν^* , en \mathbb{R}^n es por definición la aplicación

$$\nu^* : \mathcal{P}(\mathbb{R}^n) \rightarrow [0, \infty]$$

dada por

$$\nu^*(A) = \inf \left\{ \sum_{n=1}^{\infty} v(I_n) : A \subset \bigcup_{n=1}^{\infty} I_n, I_n \text{ intervalo acotado de } \mathbb{R}^n \right\} \in [0, \infty]$$

para todo $A \subset \mathbb{R}^n$ (obviamente, existen sucesiones (I_n) de intervalos acotados de \mathbb{R}^n que verifican las condiciones exigidas en la definición).

Teorema 1.1.1. (EXISTENCIA Y UNICIDAD DE LA MEDIDA DE LEBESGUE).

i) La familia de subconjuntos \mathcal{M} de \mathbb{R}^n definida como sigue

$$\mathcal{M} = \{B \cup Z : B \in \mathcal{B}, Z \subset \mathbb{R}^n \text{ con } \nu^*(Z) = 0\}$$

es una σ -álgebra en \mathbb{R}^n que contiene a los intervalos acotados.

ii) La restricción a \mathcal{M} (resp. \mathcal{B}) de la medida exterior de Lebesgue es la única medida sobre \mathcal{M} (resp. \mathcal{B}) que extiende el volumen de los intervalos acotados.

Definición 1.1.5. (MEDIDA DE LEBESGUE EN \mathbb{R}^n). La medida de Lebesgue, ν , en \mathbb{R}^n es la restricción de la medida exterior de Lebesgue a la σ -álgebra \mathcal{M} definida en el teorema anterior.

1.2. Método de la Pendiente Máxima

Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}$. Consideremos el problema de optimización

$$m = \inf [f(x) : x \in \mathbb{R}^n] \quad (P)$$

Existen diversos métodos numéricos ([1]) para solucionar este tipo de problema, aunque aquí nos centraremos en el método de la Pendiente Máxima, el cual pertenece a la familia de los métodos de descenso. Un método de descenso consiste en tomar un punto de partida $x_0 \in \mathbb{R}^n$, luego, construir una sucesión $\{x_k\} \subset \mathbb{R}^n$ tal que $f(x_k) < f(x_{k-1})$ para todo k hasta que alguna condición de optimalidad (o sea alguna condición necesaria o suficiente para que x_k sea solución del problema (P)) sea verificada. El siguiente teorema es de vital importancia en el desarrollo de esta sección.

Teorema 1.2.1. Sean $f : \mathbb{R}^n \rightarrow \mathbb{R}$ de clase C^1 y $x, d \in \mathbb{R}^n$ tales que $\langle \nabla f(x), d \rangle < 0$. Entonces existe $\delta > 0$ tal que $f(x + td) < f(x)$, $\forall t \in \langle 0, \delta \rangle$.

Prueba. Como f es de clase C^1 entonces existe $\delta > 0$ tal que $\langle \nabla f(x + td), d \rangle < 0$, $\forall t \in \langle 0, \delta \rangle$. Así, por el teorema del valor medio, para cada $t \in \langle 0, \delta \rangle$ existe $\theta \in \langle 0, t \rangle$ tal que $f(x + td) - f(x) = \langle \nabla f(x + \theta d), d \rangle < 0$. ■

La formulación general de un algoritmo de descenso viene a continuación. Supongamos que ya tenemos a x_k . Entonces:

- 1) Se determina una dirección de descenso, es decir un vector $d_k \in \mathbb{R}^n$, tal que $\langle \nabla f(x_k), d_k \rangle < 0$.
- 2) Se busca $t_k > 0$ tal que $f(x_k + t_k d_k) < f(x_k)$.
- 3) Se hace $x_{k+1} = x_k + t_k d_k$ y se repite el proceso (hasta obtener x_k óptimo o una buena aproximación).

Aunque puedan haber diversas maneras de elegir una dirección d de descenso, (y cada una de ellas dará lugar a un algoritmo de descenso distinto), la elección más natural es aquella que haga más negativo el valor $\langle \nabla f(x_k), d \rangle$. Ésta es la dirección antiparalela a la de $\nabla f(x_k)$. Entonces podemos entonces tomar $d_k = -\nabla f(x_k)$. Esta elección da lugar al Método de la Pendiente máxima (steepest descent).

Una vez elegida la dirección de descenso d_k , el Teorema 1.2.1 garantiza la posibilidad de elegir la longitud de paso t_k que cumpla los requerimientos del método. Aunque existen diversos algoritmos de búsqueda del valor óptimo de t_k , en la sección 1.3.2 tomaremos una elección práctica muy común en la elaboración del algoritmo de aprendizaje de una Red Neuronal.

1.3. Redes Neuronales Artificiales

Un modelo simplificado de una **neurona** biológica real está compuesto del **cuerpo**, un conjunto de fibras que llegan del cuerpo llamadas **dendritas**, y una fibra que sale del cuerpo llamada **axón**.

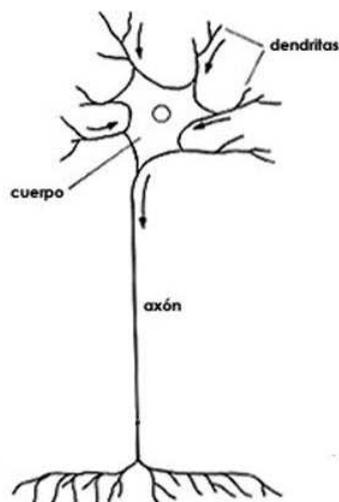


Figura 1.1: Partes principales de una neurona biológica

Las dendritas traen información en forma de impulsos eléctricos hacia el cuerpo. Por lo contrario el axón, transmite información en forma de impulsos eléctricos del cuerpo hacia otras neuronas mediante el proceso de **sinapsis**.

La transmisión de los impulsos eléctricos desde una neurona hacia la otra depende de la eficiencia de la transmisión sináptica. A esta eficiencia podemos llamarle variable **W**. Una neurona real puede variar la cantidad de impulsos por segundo, dependiendo del estado de su actividad. La recepción de información que depende del tiempo mediante las dendritas esta representado por las variables **X**. El procesamiento de la información en el cuerpo de la neurona se realiza por medio de funciones internas que generan la magnitud de salida **out**. El axón se modela mediante la variable de salida **O**.

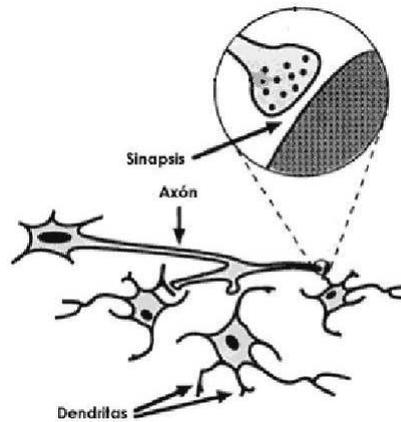


Figura 1.2: Sinapsis

Como elementos fundamentales del sistema nervioso, las neuronas están distribuidas en capas formando una red, donde esencialmente cada neurona envía su información de estado multiplicado por el correspondiente peso a todas las neuronas conectadas con ésta. Luego, cada una a su vez suma la información recibida desde sus dendritas para actualizar sus estados respectivos y seguir propagando la información .

Una de las principales características de este tipo de sistemas es su capacidad de aprendizaje. Éste es el nombre que se le da al proceso por el cual la red es expuesta a un conjunto de datos de información que ésta asimila como ejemplos para que cuando sea expuesta ante situaciones nuevas actúe de modo coherente a la información asimilada. Durante el proceso de aprendizaje cada neurona va modificando su forma de interactuar con las demás.

En esta sección daremos una formulación matemática que imitará el funcionamiento del sistema biológico neuronal descrito anteriormente, con la finalidad de aprovechar sus consecuencias prácticas en el Capítulo 4.

Empezamos denotando $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$.

Definición 1.3.1. Diremos que $Z \subset \mathbb{N} \times \mathbb{N}_0$ es un conjunto de nodos completamente indexado si se cumplen las tres condiciones siguientes:

- i) Z es finito.

ii) $(1, 1) \in Z$.

iii) $\forall (i, j) \in Z : k \in \{1, 2, \dots, i\}, l \in \{0, 1, \dots, j\} \implies (k, j), (1, l) \in Z$.

En ese caso cada elemento de Z será llamado *nodo*.

Obsevación 1.3.1. Por las condiciones i) y ii) tenemos que si Z es un conjunto de nodos completamente indexado entonces Z tiene al menos dos elementos: $(1, 0)$ y $(1, 1)$.

Ejemplo 1.3.1. $Z_1 = \{(1, 0), (1, 1)\}$ es un ejemplo de un conjunto de nodos completamente indexado, pues cada uno de sus elementos cumple la condición iii) de la definición:

. $(i, j) = (1, 0) \in Z : k \in \{1\}, l \in \{0\} \implies (k, j) = (1, 0) \in Z, (1, l) = (1, 0) \in Z$.

. $(i, j) = (1, 1) \in Z : k \in \{1\}, l \in \{0, 1\} \implies (k, j) = (1, 1) \in Z, (1, l) = (1, 0) \in Z \vee (1, l) = (1, 1) \in Z$.

Ejemplo 1.3.2. El conjunto $Z_2 = \{(1, 0), (2, 0), (4, 0), (1, 1), (2, 1), (3, 1), (1, 2), (3, 2)\}$ no es un conjunto de de nodos completamente indexado, pues no se cumple la condición iii) de la Definición 1.3.1 con $(i, j) = (4, 0)$ y $k = 3$.

Ejemplo 1.3.3. Si unimos $\{(3, 0), (2, 2)\}$ con el conjunto Z_2 del ejemplo anterior, entonces el conjunto resultante Z será un conjunto de nodos completamente indexado.

Una manera práctica de ver si un conjunto $Z \subset \mathbb{N} \times \mathbb{N}_0$ dado es un conjunto de nodos completamente indexado es hacer un arreglo de los pares ordenados que conforman el conjunto Z (cual si se tratara de los índices de una matriz) y comprobar que en cada columna no quede ningun espacio por completar. Ilustramos esto con el siguiente gráfico del conjunto del Ejemplo 1.3.2:

$(1, 0)$	$(1, 1)$	$(1, 2)$
$(2, 0)$	$(2, 1)$	
	$(3, 1)$	$(3, 2)$
$(4, 0)$		

en el que se aprecia las columnas que quedan por completar. Esto hace comprensible la elección del conjunto del Ejemplo 1.3.3 el cual si resulta ser un conjunto de nodos completamente indexado.

Nuestro siguiente paso en el camino trazado en esta sección es adicionar una manera de conectar los elementos de Z . Por eso establecemos la siguiente definición:

Definición 1.3.2. Sea Z un conjunto de nodos completamente indexado, un conjunto de conexiones en Z es un subconjunto $V \subset Z \times Z$. Dado $(i, j) \in Z$, definimos

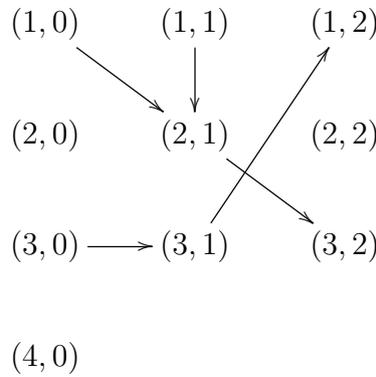
$$V_{ij} = \{(k, l) \in Z : (k, l, i, j) \in V\}$$

el cual será llamado conjunto de conexiones hacia (i, j) .

Ejemplo 1.3.4. En el conjunto Z del Ejemplo 1.3.3 definimos el conjunto de conexiones

$$V_1 = \{(1, 0, 2, 1), (3, 0, 3, 1), (3, 1, 1, 2), (2, 1, 3, 2), (1, 1, 2, 1)\}$$

Una manera gráfica de representar a un conjunto de nodos completamente indexado Z y un conjunto de conexiones en Z es trazando flechas, por ejemplo con el conjunto de conexiones V_1 del ejemplo anterior, su representación gráfica será:



Si Z es un conjunto de nodos completamente indexado, denotamos:

$$L + 1 = \max \{j \in \mathbb{N} : (1, j) \in Z\}$$

Para cada $j \in \{0, 1, \dots, L + 1\}$ denotamos

$$I_j + 1 = \max \{i \in \mathbb{N} : (i, j) \in Z\},$$

luego por la definición de Z

$$(i, j) \in Z \Leftrightarrow i \in \{1, 2, \dots, I_j + 1\}.$$

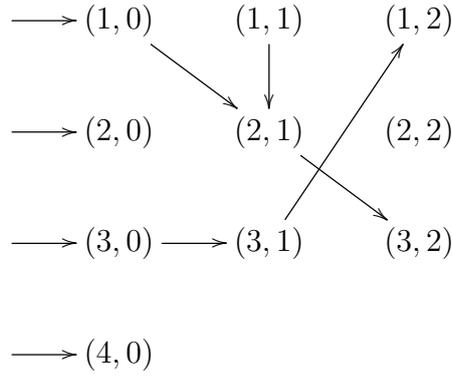
El conjunto

$$V_Z = \{(i, 0, i, 0) : i \in \{1, 2, \dots, I_0 + 1\}\}$$

será llamado conjunto de conexiones de entrada de Z . Si tomamos el conjunto Z del Ejemplo 1.3.3, entonces su conjunto de conexiones de entrada será

$$V_Z = \{(1, 0, 1, 0), (2, 0, 2, 0), (3, 0, 3, 0), (4, 0, 4, 0)\}$$

La representación gráfica del conjunto de conexiones que resulta de la unión de éste y del conjunto de conexiones del Ejemplo 1.3.4 es:



También escribimos $\mathbb{R}_{ij} = \mathbb{R}^{\text{card}V_{ij}}$. Para cada $\bar{w} \in \mathbb{R}_{ij}$ hay una biyección entre los elementos (k, l) de V_{ij} y las componentes de \bar{w} , luego, podemos denotar por w_{ij}^{kl} la (k, l) -ésima componente de \bar{w} .

Ejemplo 1.3.5. Con la misma información del Ejemplo 1.3.4, como $V_{21} = \{(1, 0); (1, 1)\}$ entonces $\text{card } V_{21} = 2$; luego, $\mathbb{R}_{21} = \mathbb{R}^2$, y cada elemento \bar{w} de \mathbb{R}_{21} tiene dos componentes: w_{21}^{10} y w_{21}^{11} .

Definición 1.3.3. (NEURONA). Sea Z un conjunto de nodos completamente indexado, V un conjunto de conexiones en Z y $(i, j) \in Z$ tal que $V_{ij} \neq \emptyset$.

1. Diremos que el par $\eta = (\bar{w}, f)$ es una (i, j) neurona si $\bar{w} \in \mathbb{R}_{ij}$ y $f : \mathbb{R} \rightarrow \mathbb{R}$ es una función.
2. El vector \bar{w} será llamado vector de pesos sinápticos y la función f será llamada función de activación.
3. Dado $\bar{x} \in \mathbb{R}_{ij}$, al valor $o = f(\langle \bar{w}, \bar{x} \rangle)$ se le denominará salida de η ; donde $\langle \cdot, \cdot \rangle$ es el producto interno usual. Así, \bar{x} recibirá el nombre de entrada de η . Siendo más explícitos, tendremos

$$o = f \left(\sum_{(k,l) \in V_{ij}} w_{ij}^{kl} x_{ij}^{kl} \right) \quad (1.1)$$

Obsevación 1.3.2. Cuando sea necesario escribiremos η_{ij} en lugar de η y o_{ij} en lugar de o para resaltar la dependencia de $(i, j) \in Z$ y simplemente diremos la neurona η_{ij} en lugar de la (i, j) -neurona η .

Definición 1.3.4. (RED NEURONAL) Una red neuronal es un triplete $\Lambda = (Z, V, \mathcal{N})$, donde Z es un conjunto de de nodos completamente indexado, V es un conjunto de conexiones en Z que contiene al conjunto de conexiones de entrada V_Z y $\mathcal{N} = \{\eta_{ij}\}_{V_{ij} \neq \emptyset}$ es un conjunto de neuronas.

Sea $\Lambda = (Z, V, \mathcal{N})$ una red neuronal, llamaremos la j -ésima capa de Λ al conjunto

$$\mathcal{N}_j = \{\eta_{ij} : (i, j) \in Z\}$$

entonces la 0-ésima y $(L + 1)$ -ésima capa de Λ serán llamadas capa de entrada y de salida de Λ respectivamente. Las capas restantes serán llamadas capas ocultas. Luego, L es el número de capas ocultas de Λ . Además nos referiremos a η_{ij} como la i -ésima neurona de la j -ésima capa de Λ .

Estamos ahora en condiciones de dar un ejemplo bastante representativo de una red neuronal en particular, asimismo, aprovecharemos su estructura para mostrar como se representa gráficamente la misma:

EL PERCEPTRÓN SIMPLE. Sea $n \in \mathbb{N}$, $\bar{w} = (w_1, w_2, \dots, w_{n+1}) \in \mathbb{R}^{n+1}$ y f una función real. El triplete $\Lambda = (Z, V, \mathcal{N})$ es llamado perceptrón simple. Donde

$$Z = \{(i, 0) : i \in \{1, 2, \dots, n + 1\}\} \cup \{(1, 1)\}$$

$$V = \{(i, 0, 1, 1) : i \in \{1, 2, \dots, n + 1\}\} \cup V_Z$$

$$\mathcal{N} = \{\eta_{ij} : (i, j) \in Z\}$$

tal que, para $(i, j) \in Z$:

$$\eta_{ij} = \begin{cases} (1, id_{\mathbb{R}}); & j = 0 \\ (\bar{w}, f); & j = 1 \end{cases} \quad (1.2)$$

donde $id_{\mathbb{R}}$ es la función identidad en \mathbb{R} .

Se puede apreciar que $V_{i0} = \{(i, 0)\}$, $\mathbb{R}_{i0} = \mathbb{R}$, $\forall i = 1, 2, \dots, n + 1$, además $V_{11} = \{(1, 0), (2, 0), \dots, (n + 1, 0)\}$ y $\mathbb{R}_{11} = \mathbb{R}^{n+1}$.

Sea $X = (X_1, X_2, \dots, X_n) \in \mathbb{R}^n$. Con la notación desarrollada hasta el momento, si colocamos $x_{i0}^{i0} = X_i \in \mathbb{R}_{i0}$, $i = 1, 2, \dots, n + 1$ donde $X_{n+1} = 1$ entonces, cada X_i servirá como entrada para la neurona (de la capa de entrada) η_{i0} , la cual producirá la misma salida $o_{i0} = id(x_{i0}^{i0}) = X_i$, si construimos el vector \bar{X} formado por cada una de estas salidas, entonces $\bar{X} = (X_1, X_2, \dots, X_n, 1) \in \mathbb{R}_{11}$, lo reescribimos como $(x_{11}^{10}, x_{11}^{20}, \dots, x_{11}^{(n+1)0})$ y podremos usarlo como entrada para esta neurona y usar su vector de pesos \bar{w} , el cual reescribimos como $(w_{11}^{10}, w_{11}^{20}, \dots, w_{11}^{(n+1)0})$ para finalmente

obtener la siguiente salida:

$$\begin{aligned}
 o_{11} &= f(\langle \bar{w}, \bar{X} \rangle) \\
 &= f\left(\sum_{(k,l) \in V_{11}} w_{11}^{kl} \cdot x_{11}^{kl}\right) \\
 &= f\left(\sum_{i=0}^{n+1} w_{11}^{i0} \cdot x_{11}^{i0}\right) \\
 &= f\left(\sum_{i=0}^n w_{11}^{i0} \cdot X_i + w_{11}^{(n+1)0}\right) \\
 &= f\left(\sum_{i=0}^n w_i \cdot X_i + w_{n+1}\right)
 \end{aligned}$$

La representación gráfica de esta red se aprecia en la Figura 1.3. Podemos extraer información útil de esta representación:

- i) Las flechas que conectan a las neuronas (excepto la que esta etiquetada con o_{11}) coinciden con las de la representación gráfica del conjunto de conexiones del conjunto de nodos completamente indexado Z definido para esta red. Asimismo, estas flechas están etiquetadas cada una con el peso sináptico correspondiente, es decir w_{ij}^{kl} representa al peso sináptico que conecta la neurona η_{kl} hacia la neurona η_{ij} , de ahí el nombre dado para V_{ij} en la Definición 1.3.2 y la notación desarrollada previamente a la Definición 1.3.3.

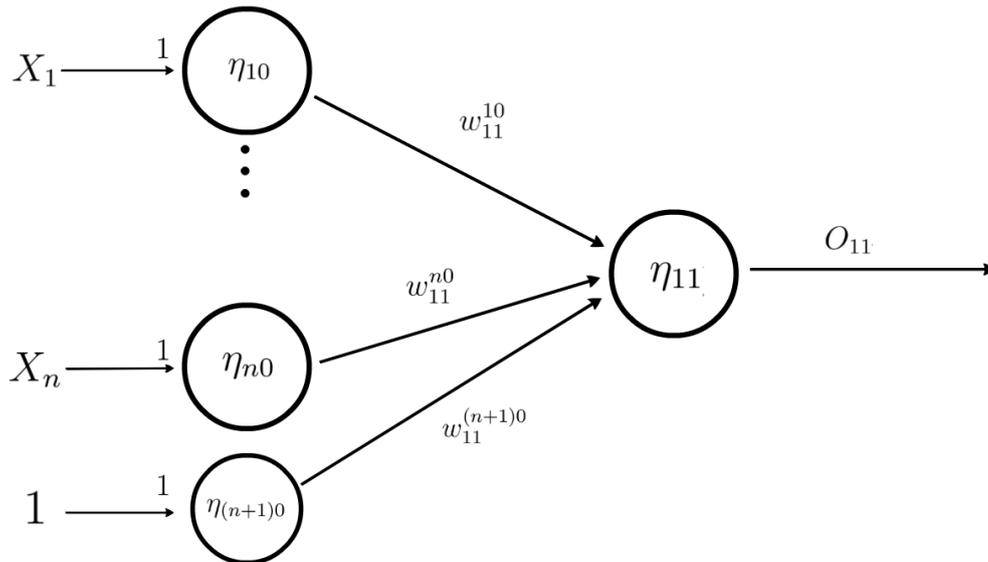


Figura 1.3: Representación gráfica del perceptrón simple

- ii) Podemos decir que el vector de entrada X es recibido por las n primeras neuronas de la capa de entrada y la última neurona de la capa de entrada recibe como entrada al escalar 1. De hecho este es el motivo por el cual hemos convenido en realizar el diagrama de el conjunto de conexiones de entrada de esa manera y también es la razón por la cual exigimos en la Definición 1.3.4 que el conjunto de conexiones V contenga al conjunto de conexiones de entrada, para que cada neurona de la capa de entrada reciba a la correspondiente componente del vector \bar{X} .
- iii) La definición de las neuronas de la capa de entrada dada en la expresión (1.2), implica que la salida de cada neurona de entrada es la misma componente de \bar{X} , de ahí que, en adelante las representaciones gráficas de las redes neuronales que usemos serán mas simplificadas y tendrán a los valores X_i , con $i = 1, 2, \dots, n+1$ insertados en los círculos de las neuronas correspondientes de la capa de entrada, reemplazando a η_{i0}
- iv) La representación gráfica de la Figura 1.3 también sugiere que la información en la red neuronal se desplaza desde la capa de entrada hacia la capa de salida (en la dirección de las conexiones), es decir hacia adelante (feedforward). En términos de lo que ya hemos definido, significa que si $(k, l) \in V_{ij}$ entonces $l < j$. Asimismo, dicha representación muestra la utilidad que tendría trabajar con este tipo de sistemas, pues podemos asimilar un conjunto de datos y transformarlos en un valor de salida que depende de los pesos sinápticos y de la función de activación.
- v) La neurona $\eta_{(n+1)0}$. recibirá el nombre de bias, y su entrada (y salida) sera igual a 1. En realidad, usaremos este nombre también cuando definamos el perceptrón multicapa, para objetos que (estrictamente hablando) de acuerdo a nuestra definición no son neuronas, pero que incluiremos en la representación gráfica de este tipo de red neuronal.

EL PERCEPTRÓN MULTICAPA. Sea $L \in \mathbb{N}$, $n_0, n_1, \dots, n_{L+1} \in \mathbb{N}$, f_1, f_2, \dots, f_{L+1} funciones reales, el conjunto

$$Z = \{(i, j) : j = 0, 1, \dots, L + 1; i = 1, 2, \dots, n_j + 1\} - \{(n_{L+1} + 1, L + 1)\} \quad (1.3)$$

es un conjunto de nodos completamente indexado, y

$$V = V_Z \cup \{(k, j-1, i, j) \in Z \times Z : j = 1, 2, \dots, L+1; \\ i = 1, 2, \dots, n_j; k = 1, 2, \dots, n_{j-1} + 1\} \quad (1.4)$$

es un conjunto de conexiones en Z que contiene al conjunto de conexiones de entrada V_Z . Por otro lado, sea el conjunto de números reales $\{w_{ij}^{kl} : (k, l, i, j) \in V\}$. Tomemos $(i, j) \in Z$ tal que $V_{ij} \neq \emptyset$, definimos el vector $\bar{w}_{ij} \in \mathbb{R}_{ij}$ cuya (k, l) -ésima componente es w_{ij}^{kl} , donde $(k, l) \in V_{ij}$. Así podemos definir:

$$\eta_{ij} = \begin{cases} (1, id_{\mathbb{R}}) & ; j = 0 \\ (\bar{w}_{ij}, f_j) & ; j = 1, 2, \dots, L+1 \end{cases} \quad (1.5)$$

y $\mathcal{N} = \{\eta_{ij}\}_{V_{ij} \neq \emptyset}$. Luego, el triplete $\Lambda = (Z, V, \mathcal{N})$ es una red neuronal artificial que recibirá el nombre de Perceptrón Multicapa.

Algunos casos particulares de la definición son los siguientes:

- . Red Cosenoidal: $L = 1$, $f_2 = id$ y $f_1 = \cos$.
- . Red Senoidal: $L = 1$, $f_2 = id$ y $f_1 = \text{sen}$.
- . Red de Fourier: $L = 1$, $f_2 = id$ y f_1 dada por

$$f_1(x) = \begin{cases} 0 & ; x \in]-\infty, -\frac{\pi}{2}] \\ \frac{1 + \cos(x + \frac{3\pi}{2})}{2} & ; x \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\ 1 & ; x \in [\frac{\pi}{2}, \infty[\end{cases} \quad (1.6)$$

- . Red Sigmoidal: $L = 1$, $f_2 = id$ y f_1 dada por la regla $f_1(x) = \frac{1}{1 + e^{-x}}$, (función sigmoide).

La representación gráfica de un perceptrón multicapa, está dada en la figura 1.4.

Notemos que según las definiciones de Z y V dadas en las expresiones (1.3) y (1.4) respectivamente, Λ tiene L capas ocultas. Para $j = 1, 2, \dots, L+1$, la j -ésima capa tiene $I_j = n_j$ neuronas y cada neurona de cada capa distinta de la capa de salida está conectada hacia cada neurona de la capa siguiente, además $V_{I_j+1j} = \emptyset$, luego, los círculos que tienen un 1 en su interior no representan a ninguna neurona en sí, sino más bien a los

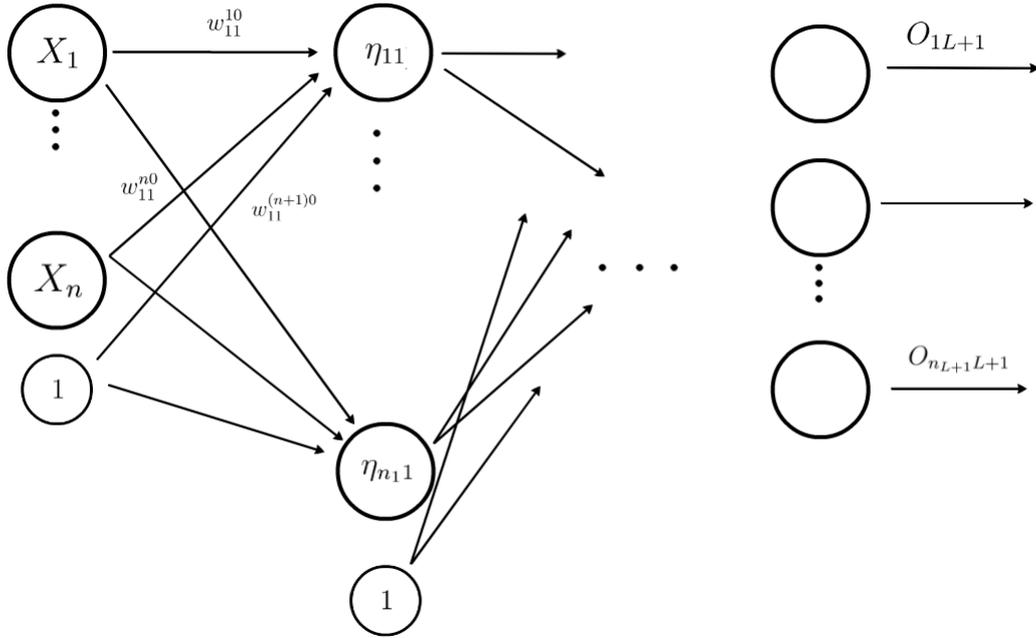


Figura 1.4: Representación gráfica del perceptrón multicapa

correspondientes nodos $(n_j + 1, j)$ que en adelante serán llamados bias y sirven para¹ sumar el peso sináptico correspondiente a la neurona de la siguiente capa con la que están conectados como si fuera una $(n_j + 1, j)$ neurona con salida igual a 1.

Entonces por convención escribiremos:

$$O_{n_j+1j} = 1, \quad j = 0, 1, \dots, L,$$

además por (1.1) y (1.4.1) se tiene

$$O_{i0} = X_i, \quad i = 1, 2, \dots, n_0,$$

luego, definimos:

$$\bar{O}_j = \begin{cases} (O_{1L+1}, O_{2L+1}, \dots, O_{n_{L+1}L+1}) & ; j = L + 1 \\ (O_{1j}, O_{2j}, \dots, O_{n_jj}, O_{n_j+1j}) & ; j \neq L + 1 \end{cases}$$

Así, para cada $j = 1, 2, \dots, L + 1$, se tiene que $\bar{O}_{j-1} \in \mathbb{R}_{ij}$ y la expresión (1.1)

¹De hecho, su importancia será apreciada en la siguiente subsección cuando veamos la capacidad de aproximación que tienen las redes neuronales, aunque en realidad dependiendo del problema a abordar, se podría prescindir de alguno de los bias en alguna capa oculta.

puede ser escrita como:

$$\begin{aligned} O_{ij} &= f_j (\langle \bar{w}_{ij}, \bar{O}_{j-1} \rangle) \\ &= f_j \left(\sum_{k=1}^{n_{j-1}+1} w_{ij}^{kj-1} O_{kj-1} \right) \end{aligned} \quad (1.7)$$

cuando $i = 1, 2, \dots, n_j$. Esta expresión será muy utilizada cuando deduzcamos las ecuaciones del algoritmo de aprendizaje del perceptrón.

La Figura 1.4 permite apreciar más claramente la similitud entre una Red Neuronal Artificial y un sistema biológico compuesto por muchas neuronas, concretamente, la fisiología del cerebro. Asimismo, muestra a una Red Neuronal Artificial como un sistema de procesadores paralelos (o sea las neuronas) conectados entre sí que transforman la información de entrada gradualmente en cada capa hasta obtener un vector que contenga la información de salida. Esta salida será entonces la evaluación de una serie de composiciones de las funciones de activación en cada capa que reciben como entradas a una combinación lineal de las salidas de la capa anterior más el valor agregado por el bias. Así, podemos usar una red neuronal para (entre otros propósitos) construir una función que transforme un conjunto de datos de entrada en otros datos de salida. Dicha función estará determinada por las funciones de activación de cada capa y los pesos sinápticos de cada conexión.

La discusión del párrafo anterior motiva las siguientes preguntas:

- a) Dada una función, ¿Cómo saber cuales son los pesos sinápticos y las funciones de activación a considerar que construyan por medio de una red neuronal una función igual o aproximada a la función dada?
- b) ¿Qué funciones pueden ser construidas a partir de una red neuronal?

Las siguientes dos subsecciones ayudan a responder dichas preguntas.

1.3.1. Capacidades de aproximación de las redes neuronales

Como hemos hecho notar en la sección anterior, una red neuronal consistente de una capa de entrada que recibe n variables y una capa de salida que produce m salidas es en realidad una representación de una función $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Nos centramos entonces en la siguiente interrogante: Si fuéramos libres de escoger cualquier vector de pesos

sinápticos \overline{W} (incluyendo los pesos asociados a los bias) y las funciones de activación f_j que deseemos, ¿qué funciones podrían ser representadas por nuestra red neuronal resultante? o mejor dicho, ¿qué tan grande sería la familia de funciones construidas de esta manera? Necesitamos algunos conceptos previos para entender mejor esta idea:

Definición 1.3.5. (DENSIDAD). Sea $K \subset \mathbb{R}^n$ un conjunto compacto, denotamos por $C(K)$ la familia de funciones continuas $f : K \rightarrow \mathbb{R}^m$. Diremos que una familia $F \subset C(K)$ es densa en $C(K)$ si y solo si para cualquier $\varepsilon > 0$ y cualquier función $g \in C(K)$ existe una función $f \in F$ tal que

$$\|g(x) - f(x)\| < \varepsilon,$$

para todo $x \in K$.

Desde que una función $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ puede ser considerada como m funciones $f_j : \mathbb{R}^n \rightarrow \mathbb{R}$, podemos suponer sin pérdida de generalidad que $m = 1$. En 1885 Weierstrass demostró que la familia de polinomios definidos sobre K es denso en $C(K)$, este resultado es conocido como el Teorema de Aproximación de Weierstrass. Muchos años después, en 1937, Stone generalizó este teorema y damos la formulación aquí:

Teorema 1.3.1. (STONE-WEIERSTRASS). Sea $K \subset \mathbb{R}^n$ un conjunto compacto y sea $F \subset C(K)$ un conjunto de funciones continuas reales sobre K satisfaciendo las siguientes condiciones:

- a) *Función constante:* La función constante $f(x) = 1$ está en F .
- b) *Separabilidad:* Para cualesquiera dos puntos $x_1 \neq x_2$ en K , existe un $f \in F$ tal que $f(x_1) \neq f(x_2)$.
- c) *Subespacio Lineal:* Para cualesquiera f y $g \in F$ y $\alpha \in \mathbb{R}$ las funciones αf y $f + g$ están en F .
- d) *Propiedad de Retículo:* Para cualesquiera f y $g \in F$, las funciones $f \vee g = \max(f, g)$ y $f \wedge g = \min(f, g)$ están en F .

Entonces F es denso en $C(K)$.

La propiedad de retículo es algo difícil de verificar. Consecuentemente una afirmación ligeramente diferente de este teorema con respecto a las propiedades de clausura algebraica es a veces más útil en aplicaciones.

Teorema 1.3.2. (STONE-WEIERSTRASS II). Sea $K \subset \mathbb{R}^n$ un conjunto compacto y sea $F \subset C(K)$ una familia de funciones continuas satisfaciendo las siguientes condiciones:

- a) *Función Constante:* La función constante $f(x) = 1$ está en F .
- b) *Separabilidad:* Para cualesquiera dos puntos $x_1 \neq x_2$ en K existe $f \in F$ tal que $f(x_1) \neq f(x_2)$.
- c) *Clausura algebraica:* Para cualesquiera $f, g \in F$ y $\alpha, \beta \in \mathbb{R}$, las funciones fg y $\alpha f + \beta g$ están en F .

Entonces F es denso en $C(K)$.

Aunque el teorema de Stone-Weierstrass tiene aplicación potencial a la aproximación de funciones continuas, muchas funciones interesantes, incluyendo funciones escalonadas son discontinuas. Esas funciones son miembros de el conjunto de funciones medibles acotadas y tienen un número finito de discontinuidades. Afortunadamente, el teorema de Stone-Weierstrass puede ser extendido a funciones medibles por medio del siguiente teorema:

Teorema 1.3.3. (LUSIN). Si g es una función real medible que es acotada en casi todo punto de un conjunto compacto $K \subset \mathbb{R}^n$, entonces para cada $\delta > 0$, existe una función real f definida en K tal que la medida del conjunto donde f no es igual a g es menor que δ , es decir:

$$\nu \{x : f(x) \neq g(x), x \in K\} < \delta$$

El Teorema 1.3.3 muestra que las funciones continuas son densas en el espacio de funciones medibles acotadas sobre un conjunto compacto K . En general para un conjunto compacto $K \subset \mathbb{R}^n$, el espacio $L^p(K)$, $1 \leq p < \infty$, es el conjunto que consiste de todas las funciones medibles Lebesgue-integrables con una norma finita, es decir:

$$L^p(K) = \{f(x) : \|f(x)\|_p < \infty, x \in K\}$$

donde la norma $\|\cdot\|_p$, $1 \leq p < \infty$, es definida como:

$$\|f\|_p = \left(\int_K |f|^p dx \right)^{1/p}$$

Por lo tanto el espacio de funciones continuas $C(K)$ en el Teorema de Stone-Weierstrass puede ser reemplazado por $L^p(K)$ así que podemos considerar no solo el problema de

aproximación usando funciones continuas sino también usando funciones discontinuas.

Para aplicaciones a redes neuronales, los primeros intentos en el estudio de aproximación usan como es natural el Teorema de Aproximación de Wierstrass y su generalización en los Teoremas de Stone Wierstrass. Así, si una red genera un espacio de funciones que satisface las condiciones del Teorema de Stone Wierstrass entonces es capaz de aproximar arbitrariamente funciones continuas definidas sobre un conjunto compacto. Esta es básicamente la idea para probar que las redes senoidales, cosenoidales y de Fourier sirven para aproximar funciones continuas. La demostración de los siguientes teoremas puede ser encontrada en [3].

Teorema 1.3.4. . Sea F el conjunto de todas las funciones que pueden ser representadas por una red cosenoidal sobre un conjunto compacto $K \subset \mathbb{R}^n$,

$$F_N = \left\{ f(x_1, x_2, \dots, x_n) = \sum_{i=1}^N u_i \cos \left(\sum_{j=1}^n w_{ij} x_j + \theta_i \right) : u_i, w_{ij}, \theta_i \in \mathbb{R} \right\}$$

$$F = \bigcup_{N=1}^{\infty} F_N$$

Entonces F es denso en $C(K)$.

Teorema 1.3.5. . Sea F el conjunto de todas las funciones que pueden ser representadas por una red senoidal sobre un conjunto compacto $K \subset \mathbb{R}^n$,

$$F_N = \left\{ f(x_1, x_2, \dots, x_n) = \sum_{i=1}^N u_i \sen \left(\sum_{j=1}^n w_{ij} x_j + \theta_i \right) : u_i, w_{ij}, \theta_i \in \mathbb{R} \right\}$$

$$F = \bigcup_{N=1}^{\infty} F_N$$

Entonces F es denso en $C(K)$.

Teorema 1.3.6. . Sea F el conjunto de todas las funciones que pueden ser representadas por una red de Fourier sobre un conjunto compacto $K \subset \mathbb{R}^n$,

$$F_N = \left\{ f(x_1, x_2, \dots, x_n) = \sum_{i=1}^N u_i f \left(\sum_{j=1}^n w_{ij} x_j + \theta_i \right) : u_i, w_{ij}, \theta_i \in \mathbb{R} \right\}$$

$$F = \bigcup_{N=1}^{\infty} F_N,$$

donde f está dada por (1.6), entonces F es denso en $C(K)$.

Es importante señalar que el hecho que una red genere un espacio de funciones definidas sobre un compacto K que satisfacen las condiciones del Teorema de Stone Weierstrass no es una condición necesaria para que dicho espacio sea denso en $C(K)$. Un ejemplo de esta afirmación lo proporciona una Red Sigmoidal, cuyo espacio de funciones generado no cumple la condición c) del Teorema 1.3.2, y sin embargo, si es denso en $C(K)$. Esto es lo que dice el siguiente teorema y cuya demostración se encuentra en ([3])

Teorema 1.3.7. . Sea F el conjunto de todas las funciones que pueden ser representadas por una red sigmoidal sobre un conjunto compacto $K \subset \mathbb{R}^n$,

$$F_N = \left\{ f(x_1, x_2, \dots, x_n) = \sum_{i=1}^N u_i f \left(\sum_{j=1}^n w_{ij} x_j + \theta_i \right) : u_i, w_{ij}, \theta_i \in \mathbb{R} \right\}$$

$$F = \bigcup_{N=1}^{\infty} F_N,$$

donde f es la función sigmoide, entonces Ω es denso en $C(S)$.

Lo anterior significa que debe haber un resultado mas general que sea equivalente al hecho de poder generar un conjunto de funciones denso en $C(K)$. Esto es lo que dice el Teorema de Aproximación Leshno (1993) ([5]), para enunciarlo damos algunos conceptos previos:

Definición 1.3.6. Decimos que una función u definida casi por doquier con respecto a la medida de Lebesgue ν sobre un conjunto medible Ω en \mathbb{R}^n es esencialmente acotada sobre Ω , lo cual denotaremos por $u \in L^\infty(\Omega)$, si $|u(x)|$ es acotada en casi todo punto de sobre Ω . Definimos la norma

$$\| u \|_{L^\infty(\Omega)} = \inf \{ \lambda : \nu \{ x : |u(x)| \geq \lambda \} = 0 \} = \operatorname{ess\,sup}_{x \in \Omega} |u(x)|.$$

Definición 1.3.7. Una función u definida casi casi por doquier con respecto a la medida de Lebesgue sobre un conjunto abierto Ω en \mathbb{R}^n es localmente esencialmente acotada sobre Ω , lo cual denotaremos por $u \in L^\infty_{loc}(\Omega)$, si para cada conjunto compacto $K \subset \Omega$, $u \in L^\infty(K)$.

Definición 1.3.8. Decimos que un conjunto F de funciones en $L^\infty_{loc}(\mathbb{R}^n)$ es denso en $C(\mathbb{R}^n)$ si para cada función $g \in C(\mathbb{R}^n)$ y para cada conjunto compacto $K \subset \mathbb{R}^n$ existe una sucesión de funciones $f_j \in F$ tal que

$$\lim_{j \rightarrow \infty} \| g - f_j \|_{L^\infty(K)} = 0.$$

Denotemos por M el conjunto de funciones que están en $L_{\text{loc}}^{\infty}(\mathbb{R})$ y tienen la siguiente propiedad: La clausura del conjunto de puntos de discontinuidad de cualquier función en M tiene medida de Lebesgue igual a cero. Esto implica que para cualquier $\sigma \in M$, intervalo $[a, b]$ y $\delta > 0$, existe un número finito de intervalos abiertos cuya unión denotamos por U , de medida δ , tal que σ es uniformemente continua en $[a, b] - U$.

Teorema 1.3.8. (LESHNO). *Sea $\sigma \in M$. Sea*

$$F_N = \left\{ f(x_1, x_2, \dots, x_n) = \sum_{i=1}^N u_i \sigma \left(\sum_{j=1}^n w_{ij} x_j + \theta_i \right) : u_i, w_{ij}, \theta_i \in \mathbb{R} \right\}$$

$$F = \bigcup_{N=1}^{\infty} F_N,$$

F es denso en $C(\mathbb{R}^n)$ si y sólo si σ no es un polinomio algebraico (casi por doquier).

En resumen, un perceptrón con una sola capa oculta y que posee una función de activación no polinomial en ésta, sería capaz de aproximar una función continua dada.

Como comentario final, la presencia del bias en el Teorema de Leshno es fundamental. Consideremos por ejemplo la red senoidal. Esta función es no polinomial, es continua, acotada y no constante. La ausencia de bias en la capa de entrada implicaría que

$$F_N = \left\{ f(x_1, x_2, \dots, x_n) = \sum_{i=1}^N u_i \text{sen} \left(\sum_{j=1}^n w_{ij} x_j \right) : u_i, w_{ij} \in \mathbb{R} \right\}$$

así, $F = \bigcup_{N=1}^{\infty} F_N$ constaría sólo de funciones impares, y no sería denso en $C(\mathbb{R}^n)$ pues el límite puntual de una sucesión de funciones impares es de nuevo una función impar.

1.3.2. Algoritmo de Aprendizaje de una Red Neuronal Artificial

Para los objetivos de este trabajo nos interesa tratar de resolver este problema: Dado un conjunto de N datos entrada-salida $\{(X_1^r, X_2^r, \dots, X_n^r - Y_1^r, Y_2^r, \dots, Y_m^r)\}_{r=1}^N$ donde cada entrada está dada por $X^r = (X_1^r, X_2^r, \dots, X_n^r)$ y cada salida es $Y^r = (Y_1^r, Y_2^r, \dots, Y_m^r)$, hallar una función que ajuste todos esos datos de acuerdo al criterio de los mínimos cuadrados y que sirva para predecir los valores futuros que tomará la variable Y . En principio, si hubiera alguna función continua $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ tal que $f(X_1^r, X_2^r, \dots, X_n^r) = (Y_1^r, Y_2^r, \dots, Y_m^r) \forall r = 1, \dots, N$; podríamos hacer una aproximación de la misma usando una red neuronal con n entradas y m salidas. Por lo visto

en el apartado anterior, podríamos usar funciones no polinomiales para las funciones de activación, quedaría pendiente averiguar qué valores de pesos sinápticos usar en cada conexión. La sucesión de pasos a realizar para la búsqueda de estos valores se denomina **algoritmo de aprendizaje** y es lo que detallamos ahora.

Primero notemos que el error cuadrático medio total entre las salidas de la red y las salidas deseadas es igual a:

$$E = \sum_{r=1}^N e_r, \quad (1.8)$$

donde

$$e_r = \frac{1}{2} \sum_{i=1}^{n_{L+1}} (O_{iL+1} - Y_i^r)^2. \quad (1.9)$$

Desde que O_{iL+1} es la salida de la última capa, depende de los pesos sinápticos de todas las capas, luego E también. Si además las funciones de activación son diferenciables, entonces podremos calcular las derivadas parciales

$$\frac{\partial E}{\partial w_{ij}^{kj-1}}$$

del error total E con respecto a cada peso sináptico w_{ij}^{kj-1} que conecta la k -ésima neurona de la capa $j - 1$ con la i -ésima neurona de la capa j .

Notar que estamos ante el problema de optimización de hallar el conjunto de pesos sinápticos que minimizen el valor de E . Podemos utilizar el método de la pendiente máxima para solucionar el problema. Entonces empezamos con un vector de pesos iniciales arbitrario $\bar{W}(0) = \{w_{ij}^{kl}(0)\}_{(k,l,i,j) \in V}$ y a partir de ahí, según el método de descenso, la dirección de descenso sería $\frac{\partial E}{\partial \bar{W}}(\bar{W}(0))$, luego, en la iteración $k+1$ tendríamos

$$\bar{W}(k+1) = \bar{W}(k) + \Delta \bar{W}(k),$$

donde

$$\Delta \bar{W}(k) = -\alpha \frac{\partial E}{\partial \bar{W}}(\bar{W}(k))$$

y α es la longitud del paso (que por simplicidad tomaremos constante en cada iteración), la cual será llamada constante de aprendizaje de la red.

De este modo, solo debemos calcular los gradientes de E en cada etapa de la iteración. Esto podemos hacerlo usando la regla de la cadena en la expresión (1.9). Empezamos derivando respecto a los pesos de la última capa w_{iL+1}^{kL} . Para esto se debe tener en

cuenta que

$$\frac{\partial O_{jL+1}}{\partial w_{iL+1}^{kL}} = 0$$

cuando $j \neq i$. Esto es porque la salida O_{jL+1} (salida de la j -ésima neurona de la $L+1$ -ésima capa) no depende de los pesos sinápticos de otras neuronas de la $L+1$ -ésima capa. Por eso, al derivar la sumatoria en (1.9) respecto a w_{iL+1}^{kL} , solo permanece el i -ésimo sumando. Es decir:

$$\frac{\partial e_r}{\partial w_{iL+1}^{kL}} = (O_{iL+1} - Y_i^r) \frac{\partial O_{iL+1}}{\partial w_{iL+1}^{kL}} \quad (1.10)$$

Por otro lado, si derivamos respecto a w_{ij}^{kj-1} la expresión (1.7) obtenemos :

$$\frac{\partial O_{ij}}{\partial w_{ij}^{kj-1}} = f'_j (\langle \bar{w}_{ij}, \bar{O}_{j-1} \rangle) O_{kj-1} \quad (1.11)$$

esta fórmula, con $j = L+1$, la reemplazamos en (1.10) para obtener:

$$\frac{\partial e_r}{\partial w_{iL+1}^{kL}} = \delta_{iL+1} O_{kL} \quad (1.12)$$

donde

$$\delta_{iL+1} = (O_{iL+1} - Y_i^r) f'_{L+1} (\langle \bar{w}_{iL+1}, \bar{O}_L \rangle). \quad (1.13)$$

Similarmente al derivar la ecuación (1.9) respecto a los pesos de la penúltima capa w_{iL}^{kL-1} usando la regla de la cadena se tiene la igualdad:

$$\frac{\partial e_r}{\partial w_{iL}^{kL-1}} = \sum_{p=1}^{n_{L+1}} \frac{\partial e_r}{\partial O_{pL+1}} \cdot \frac{\partial O_{pL+1}}{\partial w_{iL}^{kL-1}} \quad (1.14)$$

y para desdoblar cada sumando, derivamos la expresión (1.7) usando nuevamente la regla de la cadena y así, la expresión (1.14) es igual a

$$\begin{aligned} & \frac{\partial e_r}{\partial w_{iL}^{kL-1}} (O_{pL+1} - Y_p^r) f'_{L+1} (\langle \bar{w}_{pL+1}, \bar{O}_L \rangle) \left\langle \bar{w}_{pL+1}, \frac{\partial \bar{O}_L}{\partial w_{iL}^{kL-1}} \right\rangle \\ &= \sum_{p=1}^{n_{L+1}} \delta_{pL+1} \left\langle \bar{w}_{pL+1}, \frac{\partial \bar{O}_L}{\partial w_{iL}^{kL-1}} \right\rangle \\ &= \sum_{p=1}^{n_{L+1}} \delta_{pL+1} w_{pL+1}^{iL} \frac{\partial O_{iL}}{\partial w_{iL}^{kL-1}} \end{aligned} \quad (1.15)$$

Aquí usamos (1.11) con $j = L$ y reemplazamos en la última igualdad de (1.15) para obtener:

$$\frac{\partial e_r}{\partial w_{iL}^{kL-1}} = \delta_{iL} O_{kL-1} \quad (1.16)$$

donde

$$\delta_{iL} = \sum_{p=1}^{n_{L+1}} \delta_{pL+1} w_{pL+1}^{iL} f'_L(\langle \bar{w}_{iL}, O_{iL-1} \rangle)$$

Podemos continuar con este proceso retrocediendo por los pesos sinápticos de cada capa y obtendremos entonces la fórmula general de las derivadas parciales:

$$\frac{\partial e_r}{\partial w_{ij}^{kj-1}} = \delta_{ij} O_{kj-1} \quad (1.17)$$

donde

$$\delta_{ij} = \sum_{p=1}^{n_{j+1}} \delta_{pj+1} w_{pj+1}^{ij} f'_j(\langle \bar{w}_{ij}, \bar{O}_{j-1} \rangle) \quad (1.18)$$

para $j = 1, 2, \dots, L, i = 1, 2, \dots, n_j + 1$. y δ_{iL+1} está dada en la expresión (1.13).

Las expresiones (1.13), (1.17) y (1.18) constituyen la denominada regla delta, y el proceso de hallazgo de las derivadas parciales respecto de los pesos sinápticos, empezando desde la última capa hasta la primera se denomina algoritmo de retropropagación (backpropagation).

El proceso de ejecución del algoritmo de aprendizaje se denomina entrenamiento de la red. Una práctica bastante común en este proceso es dividir el conjunto de N datos entrada-salida en p grupos (llamados épocas) con la misma cantidad de elementos y aplicar un número fijo q de iteraciones en cada grupo empezando por el primer grupo y continuando hasta el último. En total se habrían realizado entonces pq iteraciones.

1.3.3. Redes Neuronales Recurrentes

Hasta el momento se sabe como hacer un modelo de predicción del tipo $Y = f(X)$ por medio de un perceptrón, en este tipo de esquema, las variables de entrada son todas independientes entre sí, y ninguna dependencia explícita del tiempo es manifestada, sin embargo, no todos los procesos naturales ocurren o pueden ser modelados bajo ese criterio. Existen procesos en los que la variable de salida depende de valores anteriores de la misma, es decir siguen un comportamiento de la forma $Y^r = f(X^r, Y^{r-1}, Y^{r-2}, \dots, Y^{r-d})$, para cierto número natural d . Para modelar tal comportamiento por medio de una red neuronal sería necesario hacer una modificación a la forma en como recorre la información a lo largo de la red ya que el valor de la variable

de salida en una determinada etapa depende de los valores de la misma en algunas etapas anteriores. Esto se muestra en la figura 1.5.²

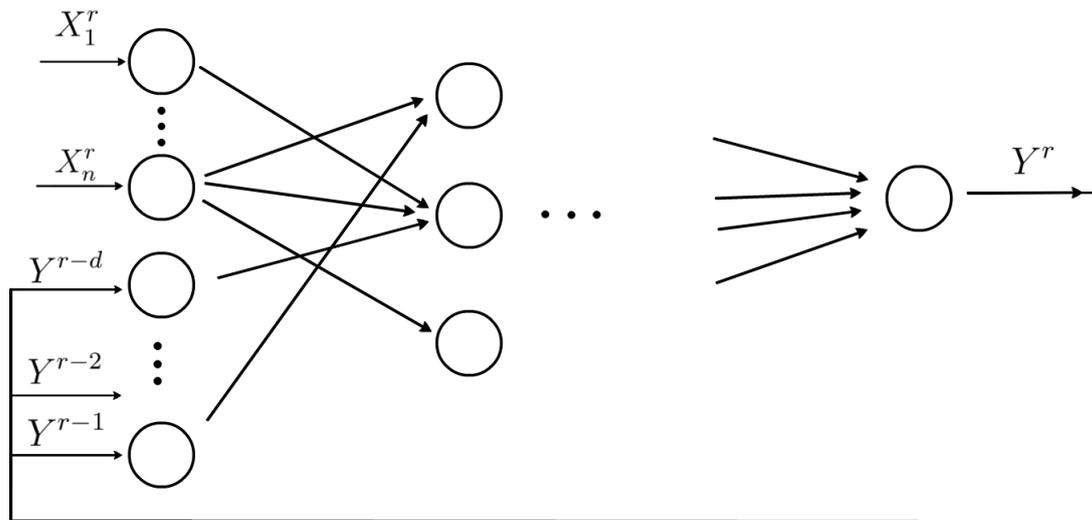


Figura 1.5: Red neuronal recurrente.

Así, como se puede ver, esta red está programada para producir una salida que regresará como entrada a la red para obtener el próximo valor de salida, este proceso se repite cada vez que se quiera obtener el siguiente valor de salida. Como consecuencia del nuevo camino que sigue la información a lo largo de la red, la dependencia del valor de salida de los pesos sinápticos sufre una modificación que se ve reflejada en las ecuaciones del gradiente del error y que desarrollamos a continuación.

Algoritmo de aprendizaje de una red neuronal recurrente.

Para ilustrar la elaboración del algoritmo de aprendizaje de una red neuronal recurrente, trabajaremos con una red cuya capa de salida tenga solo una neurona (es decir $n_{L+1} = 1$). El error E está dado por

$$E = \sum_{r=1}^N e_r,$$

²Aunque también podríamos hacer un apartado especial para definir formalmente una red neuronal recurrente, por simplicidad, con la representación gráfica de la misma y el conocimiento de cada una de las funciones de activación y los pesos sinápticos será suficiente para mostrarla completamente. Lo mismo se hará para las redes neuronales que trabajaremos en adelante. De hecho, es de esa forma en que se definen las redes neuronales en la mayoría de textos sobre el tema.

donde

$$e_r = \frac{(O_{1L+1} - Y^r)^2}{2}, \quad (1.19)$$

habiendo sido ya provistos de un conjunto de patrones de aprendizaje

$$\{(X_1^r, \dots, X_n^r, Y^{r-1}, \dots, Y^{r-d}) - (Y^r)\}_{r=1}^N \quad (1.20)$$

donde d es el orden de retraso.

Notar que al derivar la expresión (1.19) respecto a w_{ij}^{kl} se tiene

$$\frac{\partial e_r}{\partial w_{ij}^{kl}} = (O_{1L+1} - Y^r) \frac{\partial O_{1L+1}}{\partial w_{ij}^{kl}} \quad (1.21)$$

Para calcular el segundo factor del producto en la igualdad anterior, usamos la regla de la cadena, aquí es donde surge la diferencia en el cálculo de los gradientes respecto al algoritmo backpropagation desarrollado en la subsección anterior pues la salida O_{1L+1} en la etapa r , que en adelante denotaremos por $y(r)$, además de depender de los pesos sinápticos w_{ij}^{kl} , depende de los valores de las salidas O_{1L+1} en las etapas anteriores $r-1, r-2, \dots, r-d$, es decir de $y(r-1), y(r-2), \dots, y(r-d)$ respectivamente, las cuales a su vez dependen de los pesos sinápticos w_{ij}^{kl} . Así, si denotamos por $\bar{y} = (y(r-1), y(r-2), \dots, y(r-d))$, y por \bar{W} al vector de pesos sinápticos, entonces tenemos una relación del tipo

$$y(r) = G(\bar{W}, \bar{y}) = G(\bar{W}, \bar{y}(\bar{W}))$$

luego,

$$\frac{\partial y(r)}{\partial \bar{W}} = \frac{\partial G}{\partial \bar{W}} + \frac{\partial G}{\partial \bar{y}} \frac{\partial \bar{y}}{\partial \bar{W}}$$

el segundo sumando en la igualdad anterior se debe a lo mencionado anteriormente, es decir, la dependencia de \bar{y} respecto a \bar{W} . La expresión anterior será escrita más convenientemente como

$$\frac{\partial y(r)}{\partial \bar{W}} = \frac{\partial^e y(r)}{\partial \bar{W}} + \frac{\partial^e y(r)}{\partial \bar{y}} \frac{\partial \bar{y}}{\partial \bar{W}}. \quad (1.22)$$

El símbolo $\frac{\partial^e}{\partial \bar{W}}$ representa la derivada explícita usual respecto a la primera variable, el significado es análogo para $\frac{\partial^e}{\partial \bar{y}}$. Con esta aclaración, queda clara la diferencia entre $\frac{\partial y(r)}{\partial \bar{W}}$ y $\frac{\partial^e y(r)}{\partial \bar{W}}$ pues la primera expresión denota la derivada total de la salida $y(r)$ (incluyendo la dependencia explícita e implícita) respecto a \bar{W} . Finalmente, para hallar $\frac{\partial e_r}{\partial w_{ij}^{kl}}$ tenemos en cuenta que la ecuación (1.21) es igual a:

$$\frac{\partial e_r}{\partial w_{ij}^{kl}} = (y(r) - Y^r) \frac{\partial y(r)}{\partial w_{ij}^{kl}} \quad (1.23)$$

donde, por la definición de \bar{y} , la expresión (1.22) puede ser escrita como:

$$\frac{\partial y(r)}{\partial w_{ij}^{kl}} = \frac{\partial^e y(r)}{\partial w_{ij}^{kl}} + \sum_{s=1}^d \frac{\partial^e y(r)}{\partial y(r-s)} \frac{\partial y(r-s)}{\partial w_{ij}^{kl}}. \quad (1.24)$$

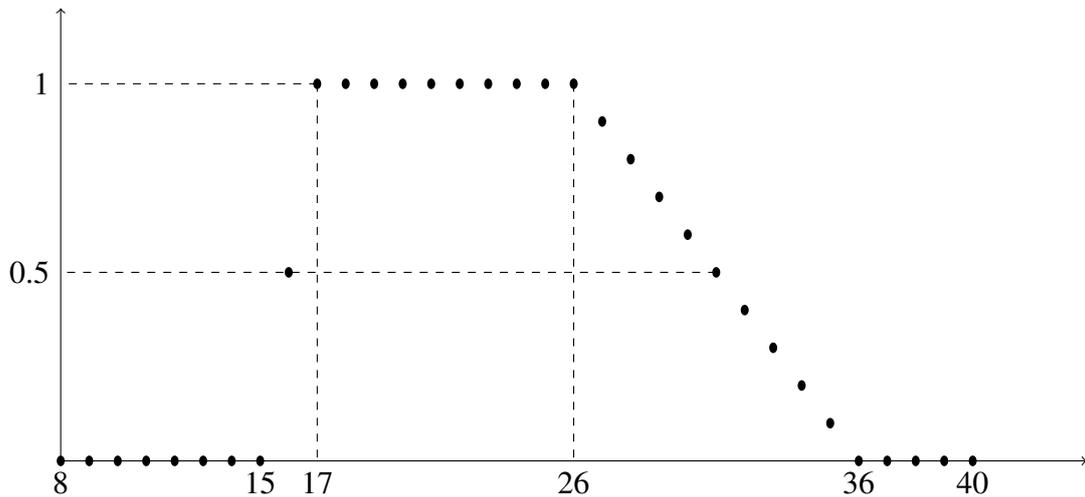
La ecuación anterior es la fórmula que sirve para hallar el gradiente de la función de error y aplicar luego el método de la pendiente máxima. Los términos $\frac{\partial^e y(r)}{\partial w_{ij}^{kl}}$ pueden ser calculados mediante el algoritmo backpropagation usual.

1.4. Lógica Difusa

1.4.1. Conjuntos Difusos

A menudo la clase de objetos encontrados en el mundo real no tienen un criterio estrictamente definido de pertenencia a ciertos conjuntos. Por ejemplo, la clase de los animales claramente incluye a perros, caballos, pajaros, etc. como sus miembros y claramente excluye a otros objetos como rocas, fluidos, plantas, etc. Sin embargo tales objetos como estrellas de mar, bacterias, etc. ocupan una posición ambigua con respecto a la clase de los animales, pues no podemos decir con seguridad que dichos objetos son animales. El mismo tipo de ambigüedad surge en el caso de la *clase de los hombres jóvenes*, el cual no constituye estrictamente a un conjunto, en el sentido matemático ya que no existe una edad límite precisa establecida como criterio para diferenciar entre un hombre joven y un hombre que no lo es. Así, la diferencia entre los enunciados “*La edad de David es 28 años*” y “*David es un hombre joven*” es que el primero tiene información más precisa, a su vez constituye formalmente una proposición, es decir, podemos asegurar la verdad o falsedad del mismo; a diferencia del segundo, sobre el cual, lo más que podemos analizar usando el lenguaje humano común (e impreciso) es *qué tan verdadero* o *qué tan falso* es. En ese sentido el adjetivo *joven* puede ser reconocido como un *valor lingüístico* de la variable *edad* entendiendo que juega el mismo rol que el valor numérico 28, sólo que menos preciso y menos informativo. Lo mismo es verdad para los valores lingüísticos *muy joven*, *extremadamente joven*, *no tan joven*, *no joven*, *no muy joven*, etc.

En un intento de estudiar con más formalidad los comentarios del párrafo anterior, consideremos el conjunto $X = \{8, 9, 10, \dots, 40\}$ (un determinado rango de edades) y a cada elemento de X asociemos un número entre 0 y 1 para formar el conjunto J de pares dibujados en el siguiente gráfico:



Para cada par (x, y) presente en J , podemos interpretar el valor y como un número que expresa *qué tan verdadero* resulta decir que *un hombre que tiene la edad x es joven*. Así, si David tiene 28 años, podemos dar al enunciado “*David es un hombre joven*” un grado de cumplimiento de 0,8, o decir que esta proposición (ahora en este sentido más general) tiene un *grado de veracidad del 80 %*.

Cada par del conjunto J no ha sido escogido al azar, de acuerdo al significado del término lingüístico *joven*, mientras más edad tenga una persona, menos derecho tiene de recibir este adjetivo, de ahí la forma del gráfico de los pares en el plano. De la misma manera debería comportarse un conjunto de pares que cumpla un rol análogo para los términos lingüísticos *muy joven*, *extremadamente joven*, etc. Este es el primer paso en nuestro objetivo de dar un tratamiento más preciso a las proposiciones ambiguas e imprecisas del lenguaje humano común y que inspiran las definiciones dadas a lo largo de toda esta sección.

Definición 1.4.1. Sea X un espacio de objetos (universo) y $f : X \rightarrow [0, 1]$ una función. Diremos que $A = \{(x, f(x)) : x \in X\}$ es un conjunto difuso y que f es una función de pertenencia de A . Para cada $x \in X$, el valor $f(x)$ será llamado el grado de pertenencia de x a A .

Definición 1.4.2. Con los mismos elementos de la definición anterior, en caso la función f sea continua, se define el centro de gravedad del conjunto difuso A como:

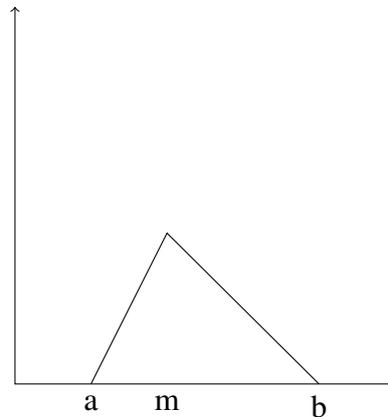
$$CG = \frac{\int_X x f(x) dx}{\int_X f(x) dx}. \quad (1.25)$$

Obsevación 1.4.1. En adelante denotaremos por f_A la función de pertenencia de A . Además, diremos que mientras más cercano a 1 sea el valor de $f_A(x)$, mayor será el grado de pertenencia de x a A .

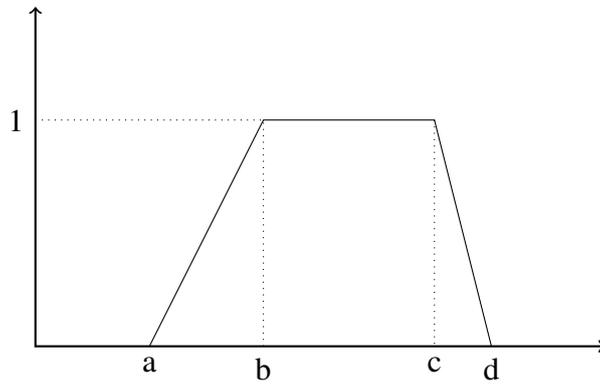
Obsevación 1.4.2. Debido a que un conjunto difuso A es únicamente determinado por su función de pertenencia f_A , por abuso de notación, diremos que f_A es un conjunto difuso. También se deduce que dos conjuntos difusos son iguales si y solo si tienen la misma función de pertenencia.

Ejemplos muy comunes de funciones de pertenencia son las funciones trapezoidales, lineales y curvas. (En todos los ejemplos siguientes, se considera $f_A : \mathbb{R} \rightarrow [0, 1]$).

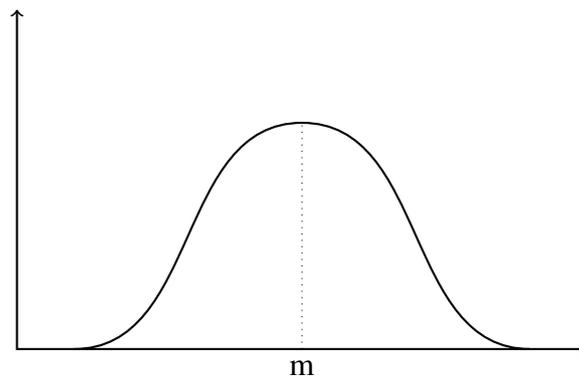
$$f_A(x) = \begin{cases} 0 & x \leq a \\ \frac{x-a}{m-a} & a < x \leq m \\ 0 & x \geq b \\ \frac{b-x}{b-m} & m < x < b \end{cases}$$



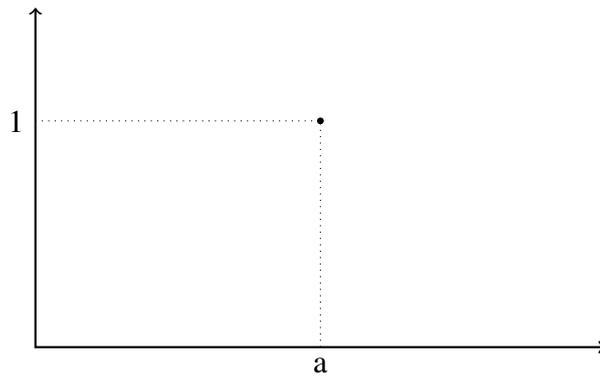
$$f_A(x) = \begin{cases} 0 & x \leq a \vee x > d \\ \frac{x-a}{b-a} & a < x \leq b \\ 1 & m \leq x < b \\ \frac{d-x}{d-c} & c \leq b \leq d \end{cases}$$



$$f_A(x) = \exp[-k(x - m)^2]$$



$$f_A(x) = \begin{cases} 0 & x \neq a \\ 1 & x = a \end{cases}$$



Definición 1.4.3. Sea X un espacio de objetos (universo) y $f_A, f_B : X \rightarrow [0, 1]$ las funciones de pertenencia asociadas a los conjuntos difusos A y B :

- i) El complemento de A , denotado por A' es el conjunto difuso $f_{A'} = 1 - f_A$.
- ii) Decimos que A está incluido en B (o equivalentemente A es un subconjunto de B), lo cual denotamos $A \subset B$, si y solo si $f_A \leq f_B$.

- iii) La unión de A y B , es un conjunto difuso C , denotado por $C = A \cup B$, con función de pertenencia $f_C = \text{Max}[f_A, f_B]$.
- iv) La intersección de A y B , es un conjunto difuso C , denotado por $C = A \cap B$, con función de pertenencia $f_C = \text{Min}[f_A, f_B]$.
- v) Un conjunto difuso es vacío si y solo si su función de pertenencia es la función nula.

La Definición (1.4.1) generaliza la noción de un conjunto en el sentido usual, pues para $A \subset X$, la función característica $\varphi_A : X \rightarrow [0, 1]$ con regla de correspondencia

$$\varphi_A(x) = \begin{cases} 0 & ; x \notin A \\ 1 & ; x \in A \end{cases}$$

es un conjunto difuso. Por otro lado, la definición (1.4.3) generaliza las operaciones entre conjuntos usuales.

En la figura siguiente se muestra las gráficas de f_A y f_B que representan a los conjuntos difusos A y B respectivamente. f_A es la función triangular. La intersección de los conjuntos difusos A y B , está representado por la curva más oscura, mientras que la parte restante representa a la unión difusa $A \cup B$.

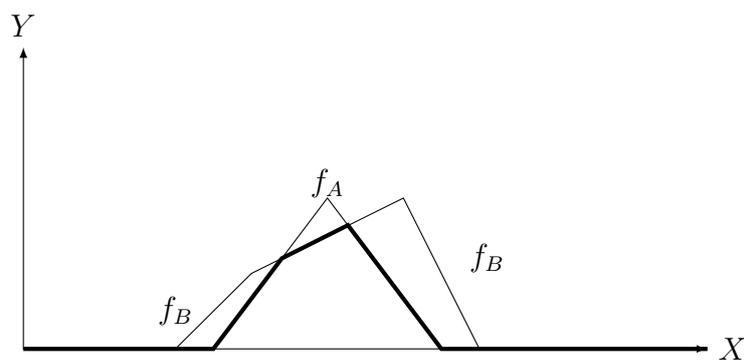


Ilustración de las gráficas de la unión e intersección

Como consecuencia de la definición, las operaciones de unión y de intersección de conjuntos difusos tienen la propiedad de asociatividad, es decir:

$$A \cup (B \cup C) = (A \cup B) \cup C,$$

$$A \cap (B \cap C) = (A \cap B) \cap C.$$

Para demostrar las igualdades anteriores basta con observar que para cada $x \in X$,

$$\begin{aligned} \text{máx} \{f_A(x), \text{máx} \{f_B(x), f_C(x)\}\} &= \text{máx} \{\text{máx} \{f_A(x), f_B(x)\}, f_C(x)\} \\ \text{mín} \{f_A(x), \text{mín} \{f_B(x), f_C(x)\}\} &= \text{mín} \{\text{mín} \{f_A(x), f_B(x)\}, f_C(x)\}. \end{aligned}$$

De forma similar se demuestra las siguientes propiedades que son análogas a las correspondientes a conjuntos ordinarios:

$$\left. \begin{aligned} (A \cup B)' &= A' \cap B' \\ (A \cap B)' &= A' \cup B' \end{aligned} \right\} \text{Leyes de Morgan.}$$

$$\left. \begin{aligned} C \cap (A \cup B) &= (C \cap A) \cup (C \cap B) \\ C \cup (A \cap B) &= (C \cup A) \cap (C \cup B) \end{aligned} \right\} \text{Leyes Distributivas.}$$

Como comentario final, notar que la noción de pertenencia, la cual juega un rol fundamental en el caso de conjuntos ordinarios, no tiene el mismo rol en el caso de conjuntos difusos. Así, no tiene sentido hablar de un punto x perteneciendo a un conjunto difuso A excepto en el caso en el que $f_A(x)$ sea positivo. Más aún, podemos introducir dos niveles α y β tales que $0 < \beta < \alpha < 1$ y convenir en que (1) x pertenece a A si $f_A(x) \geq \alpha$, (2) x no pertenece a A si $f_A(x) \leq \beta$ y (3) x tiene un estatus indeterminado respecto a A si $\beta < f_A(x) < \alpha$. Esto nos conduce a la Lógica trivaluada con tres valores de verdad: $V(f_A(x) \geq \alpha)$, $F(f_A(x) \leq \beta)$ e $I(\beta < f_A(x) < \alpha)$; significando en cada caso Verdadero, Falso e Indeterminado, respectivamente. Un tratamiento más detallado sobre este punto puede ser encontrado en ([7] y [8])

1.4.2. Inferencia Difusa

Inferencia difusa es el proceso de formulación de una función de algún valor de entrada dado hacia alguna salida usando lógica difusa. Es un sistema lógico que formaliza el razonamiento aproximado y que nace a partir de la generalización de la ley Modus Ponens de la lógica usual; la cual enuncia que cuando $v(P) = 1$ y $v(P \rightarrow Q) = 1$ entonces $v(Q) = 1$, donde $v(P)$ es el valor de verdad de la proposición P , el cual es igual a 1 si P es verdadero e igual a 0 si P es falso³. En la lógica difusa este valor podría ser el grado de cumplimiento de P el cual es un número entre 0 y 1. En este trabajo se dará una manera práctica de aplicar las ideas expuestas hasta el momento para la formulación del proceso de inferencia. El proceso detallado de la construcción de la inferencia difusa

³En términos simples quiere decir que a partir del cumplimiento de las proposiciones P y $P \rightarrow Q$ se infiere el cumplimiento de la proposición Q .

y la justificación de las definiciones que usaremos en esta parte pueden ser encontradas en ([8]).

Reglas SI-ENTONCES:

Son expresiones de la forma:

$$\text{SI } x \text{ es } A \text{ ENTONCES } y \text{ es } B.$$

donde A y B son términos lingüísticos definidos por conjuntos difusos. La parte SI (x es A) es llamado antecedente o premisa, mientras que la parte ENTONCES (y es B) es llamado consecuente o conclusión. Las siguientes expresiones constituyen algunos ejemplos simples de este tipo de reglas:

- a) SI la distancia es *cercana* ENTONCES la aceleración es *negativa*.
- b) SI el precio es *económico* entonces *la propina es la quinta parte del precio*.
- c) SI el ángulo de inclinación es *cero* y la distancia al objetivo es *cercana* ENTONCES la potencia de la grúa es *pequeña*.

En el último ejemplo se observa que el antecedente de la regla tiene la forma “ P y Q ”, donde P es: el ángulo de inclinación es *cero* y Q es: la distancia al objetivo es *cerca*. Esto ejemplifica el hecho que el antecedente puede ser resultado de una operación difusa entre dos proposiciones. Esto se describe a continuación:

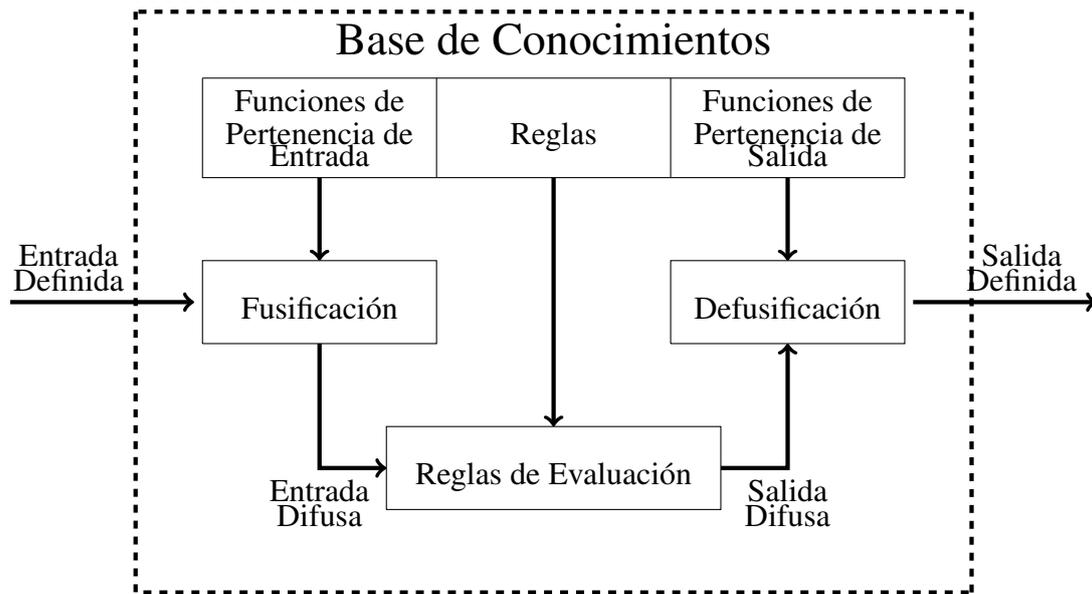
Operadores Difusos. Se definen los operadores difusos O (OR) e Y (AND). De la siguiente manera:

$$O(a, b) = \text{máx}(a, b)$$

$$Y(a, b) = \text{mín}(a, b)$$

Sirven para asociar un valor numérico concreto entre 0 y 1 al antecedente de una regla que tenga la forma “ P o Q ” o “ P y Q ” respectivamente. En este caso los valores de a y b son los grados de cumplimiento de P y Q respectivamente y son determinados por la funciones de pertenencia de los conjuntos difusos A y B . Al resultado se le llama nivel de activación de la parte SI de la regla.

Sistema de Inferencia Difuso. Un sistema de inferencia difuso se utiliza cuando la complejidad del proceso en cuestión es muy alta y no existen modelos matemáticos precisos, para procesos altamente no lineales y cuando se involucran definiciones y conocimiento no estrictamente definido (impreciso o subjetivo). En cambio, no es una buena idea usarlo cuando algún modelo matemático ya soluciona eficientemente el problema, cuando los problemas son lineales o cuando no tienen solución. El proceso se realiza en varias etapas y es mostrado en el siguiente diagrama:



A continuación se detalla cada una de las etapas del proceso de inferencia difusa:

1. **Fusificación de las entradas.** En esta etapa se toman los datos de entrada y se determina el grado en que pertenecen a cada uno de los conjuntos difusos a través de las funciones de pertenencia.
2. **Reglas Difusas** Se establecen el conjunto de reglas SI-ENTONCES, éstas pueden ser de los siguientes tipos:
 - a) **Reglas de tipo Mamdani.** El formato de estas reglas es: SI x_1 es A_1 y/o x_2 es A_2 y/o \dots y/o x_n es A_n ENTONCES y es B , donde los x_i y y son valores lingüísticos (términos lingüísticos asociados a conjuntos difusos) que las variables de entrada pueden asumir. Los sistemas que usan este tipo de reglas fueron propuestos por Mamdani en ([10]).
 - b) **Reglas de tipo Sugeno.** El formato para este tipo de reglas es: SI x_1 es A_1 y/o x_2 es A_2 y/o \dots y/o x_n es A_n ENTONCES $v = f(x_1, x_2, \dots, x_n)$ donde los x_i son las variables lingüísticas de entrada, los A_i son los valores

lingüísticos que dichas variables pueden asumir, v es la variable de salida y f representa una función lineal de las entradas. A los sistemas que usan este tipo de reglas se les conoce como sistemas TSK (Takagi-Sugeno-Kang) ([11], [12]), en honor a sus creadores.

3. **Mecanismo de Inferencia.** Esta parte constituye el núcleo del sistema de inferencia. Aquí, primero se determinan las reglas activadas por los valores de entrada y se efectúa el cálculo con dichas reglas:

a) Inferencia en sistemas de tipo Mamdani:

a) Cálculo de la parte SI de las reglas.

Se calcula el valor numérico (nivel de activación) asociado a cada antecedente de cada regla activada mediante el uso de los operadores difusos definidos anteriormente y de acuerdo a la estructura de los antecedentes cada regla.

b) Cálculo de la parte ENTONCES de las reglas.

A partir del nivel de activación se determina la conclusión de la regla mediante implicación difusa:

Implicación difusa

Sean los conjuntos difusos A y B en los universos de discurso X e Y , con funciones de pertenencia f_A y f_B respectivamente, una implicación difusa de A en B , es un conjunto difuso denotado por $A \rightarrow B$ con función de pertenencia $f_{A \rightarrow B}$ que se determina con:

. Implicación Zadeh. Se implementa de la siguiente manera:

$$f_{A \rightarrow B}(x, y) = \max \{1 - f_A(x), \min \{f_A(x), f_B(y)\}\}$$

. Implicación Mamdani. El método de Mamdani es el más usado de las implicaciones difusas y se implementa con el mínimo:

$$f_{A \rightarrow B}(x, y) = \min \{f_A(x), f_B(y)\}$$

. Implicación Larsen. Se implementa mediante el producto:

$$f_{A \rightarrow B}(x, y) = f_A(x)f_B(y)$$

Aunque hay muchas implicaciones difusas, las de Mamdani y Larsen son las de más fácil implementación y las más usadas.

c) Agregación:

Supongamos que se tiene la base de reglas:

Regla 1:

SI x_1 es A_1^1 y/o x_2 es A_2^1, \dots y/o x_m es A_m^1 ENTONCES y es B_1 .

Regla 2:

SI x_1 es A_1^2 y/o x_2 es A_2^2, \dots y/o x_m es A_m^2 ENTONCES y es B_2 .

⋮

Regla n :

SI x_1 es A_1^n y/o x_2 es A_2^n, \dots y/o x_m es A_m^n ENTONCES y es B_n .

En cada regla los subíndices j en los consecuentes B_j se refieren al número de la regla, y los subíndices i en los antecedentes A_i^j , se refieren a los conjuntos difusos asociados a los términos lingüísticos de las variables x_i .

En cada regla se evalúa primero su antecedente y luego su consecuente, según lo indicado en las partes a) y b). Se obtienen n conjuntos difusos B'_1, B'_2, \dots, B'_n como resultado de la evaluación de las n reglas difusas y del método de implicación aplicado. En caso dos o más de los n conjuntos difusos correspondan al mismo término lingüístico, el valor del lado derecho para dicho término se obtiene aplicando el operador máximo a los valores de activación. Finalmente el conjunto difuso global B' de salida está dado por:

$$B' = B'_1 \cup B'_2 \cup \dots \cup B'_n \quad (1.26)$$

En general siempre se cumple que $B'_j \subset B_j$. Recordar que la unión usada en (1.26) se define como el máximo.

b) **Inferencia en sistemas de tipo Sugeno.** En lugar de trabajar con una salida difusa, Takagi, Sugeno y Kang ([11]) propusieron un nuevo modelo basado en reglas donde el antecedente está compuesto por variables lingüísticas y el consecuente se representa como una función lineal de las variables de entrada.

a) Cálculo de la parte SI de las reglas:

El cálculo del lado izquierdo de las reglas difusas en estos sistemas es el mismo que en los sistemas Mamdani. Al aplicar el operador correspondiente a la estructura de la parte SI de la regla R_j se obtiene un grado de activación α_j .

b) Cálculo de la parte ENTONCES de las reglas:

En el lado derecho de estas reglas se obtiene el respectivo valor de salida mediante la combinación lineal de las entradas:

$$v_j = f(u_1, u_2, \dots, u_m)$$

donde el subíndice j se refiere al número de la regla activada.

c) Salida en sistemas Sugeno.

La salida de un sistema difuso TSK que usa una base de conocimiento con n reglas se obtiene como la media ponderada de las salidas individuales $v_j (j = 1, \dots, n)$ proporcionadas por las reglas activadas como sigue:

En esta expresión $v_j = f_j(u_1, u_2, \dots, u_m)$ es la salida de cada regla difusa activada y α_j es el nivel de activación o disparo que resulta de la inferencia para cada una de ellas:

$$\frac{\sum_{j=1}^n \alpha_j f_j(u_1, u_2, \dots, u_n)}{\sum_{j=1}^n \alpha_j}$$

4. **Defusificación.** La defusificación es el proceso de convertir los valores difusos de las variables de salida que resultan del proceso de inferencia en información precisa expresada mediante un valor concreto. Esta etapa es necesaria en los sistemas Mamdani, porque el conjunto difuso de salida para estos sistemas no es directamente utilizable para dar una información precisa al operador o mandar un accionador, es necesario pasar del mundo difuso al mundo real.

Hay una diversidad de métodos de defusificación, aquí presentamos tres de los más usados, los denominados Media de Alturas, Media de Máximos y Centro de Gravedad. Antes de entrar en la descripción de cada uno de ellos, introducimos algo de notación. Supongamos que como resultado del proceso de inferencia

han sido activadas n reglas. Cada una de las j -ésimas reglas ha producido una conclusión en la forma de conjunto difuso acotado por el nivel de activación α_j . Denotamos:

n = número de reglas activadas.

α_j = Nivel de activación de la j -ésima conclusión (altura).

$cmax_j$ = Valor escalar correspondiente al conjunto difuso B_j tal que $f_{B_j}(cmax_j) = 1$.

$cmax^j$ = Valor escalar correspondiente al conjunto difuso B'_j tal que $f_{B'_j}(cmax^j) = \alpha_j$.

c^* = Valor escalar concreto defusificado de la variable control.

a) Media de Alturas (MH).

Para cada regla activada R_j , se considera el $cmax_i$ correspondiente a la conclusión de la regla. La contribución de cada regla al valor de c^* es ponderada mediante el valor α_j (altura) de la j -ésima conclusión, de la siguiente manera:

$$c^* = \frac{\sum_{j=1}^n cmax_j \cdot \alpha_j}{\sum_{j=1}^n \alpha_j}.$$

En caso de para algún j , no se tenga un único $cmax_j$ sino que más bien un intervalo, como por ejemplo en una función trapezoidal, puede tomarse el valor medio del intervalo.

b) Media de Máximos (MOM).

Es el promedio de los valores $cmax^j$, es decir:

$$c^* = \frac{1}{n} \sum_{j=1}^n cmax^j.$$

c) Centro de Gravedad (CG).

Después de la fase de agregación, se obtiene un conjunto difuso B definido por la expresión (1.26). El valor de la salida defusificada será entonces igual al centro de gravedad de este conjunto difuso, valor que se calcula según (1.25).

A continuación se muestra el uso de la inferencia difusa en un caso especial.

1.4.3. Ejemplo de aplicación: Uso de Inferencia difusa para diseñar un sistema de control difuso de una grúa. ([9])

En ésta sección se muestra como diseñar un controlador difuso de una grúa pórtico para contenedores:

Para descargar contenedores desde/hacia los barcos, en la mayoría de los puertos se usan grúas pórticos. A través de cables flexibles montados en su cabezal una grúa levanta un container y luego el carro de la grúa se desplaza sobre unos rieles, trasladando el ahora container colgante hasta el lugar de descarga (ver figura 1.6). En el otro extremo del pórtico está la cabina de comando y control. Por razones de costo las acciones de cargar y descargar barcos se debe hacer en el mínimo tiempo posible.

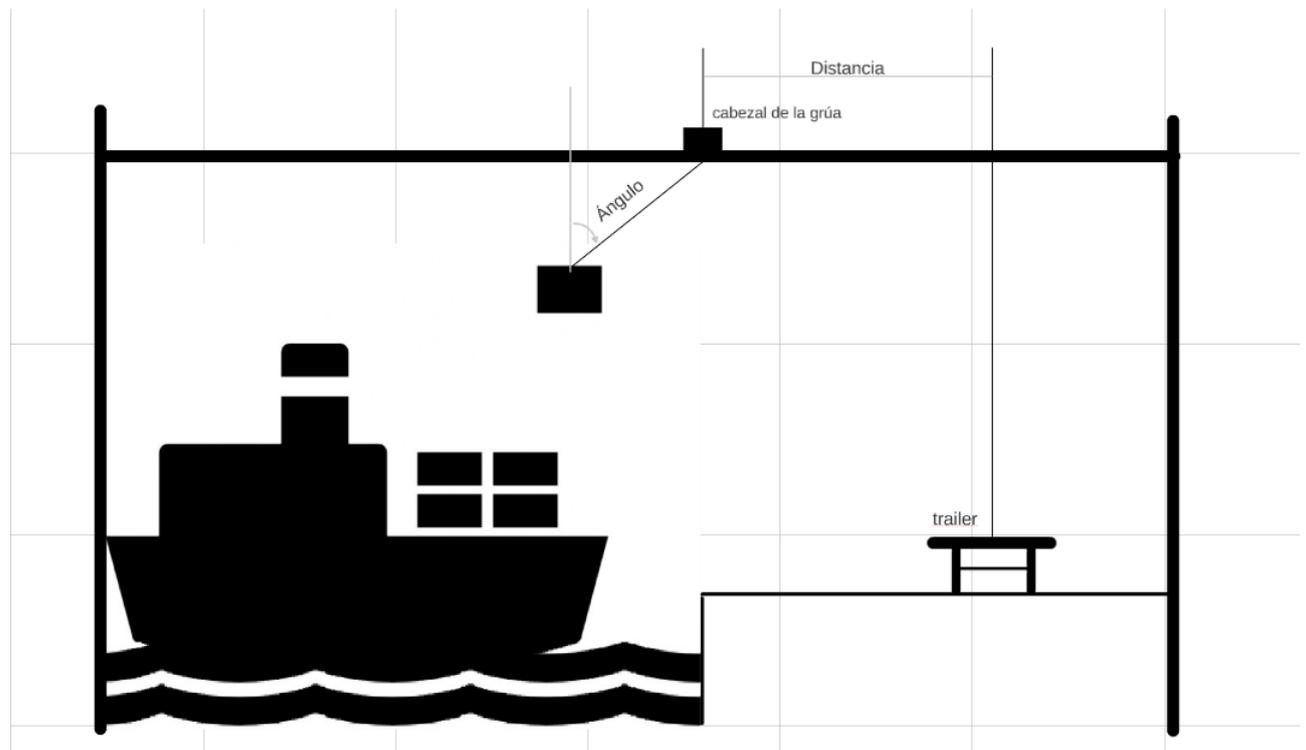


Figura 1.6: Descarga de contenedores

Cuando se levanta un contenedor y se inicia el movimiento del carro, el contenedor comienza a balancearse, pero eso no constituye un problema serio durante el desplazamiento debido a que está fuertemente sujeto, impidiendo algún posible desprendimiento. El problema de control surge cuando ya estando sobre el lugar de descarga se debe esperar a que el balanceo del container disminuya lo suficiente como para descargarlo.

Otra posibilidad es mover suavemente el container en su movimiento vertical de modo que minimice el balanceo, pero ello constituye también un retraso en el tiempo de realización de la tarea. Abordaremos el problema asumiendo los siguientes valores de entrada para el sistema:

Ángulo=40°, Distancia=2 metros.

A continuación presentamos de forma resumida las fases de análisis y diseño del sistema de inferencia difuso que usaremos para resolver el problema planteado. A partir de la descripción del problema podemos definir claramente que las variables de entrada son el **ángulo** entre el contenedor y el cabezal de la grúa, y la **distancia** entre el cabezal de la grúa y la posición meta; mientras que la variable de salida es la **potencia** del motor de la grúa.

El sistema de control automatizado de la grúa dispondrá de sensores especializados para determinar la posición del cabezal de la grúa y para el ángulo de oscilación del contenedor. El controlador difuso proporcionará la potencia del motor de la grúa requerida para desplazar el contenedor desde su posición inicial hasta su posición meta en el menor tiempo posible.

Ahora definimos los términos lingüísticos para cada variable:

a) Ángulo: Universo de discurso: $[-90^\circ, 90^\circ]$

Conjuntos difusos: positivo grande, positivo pequeño, cero, negativo pequeño, negativo grande.

b) Distancia: Universo de discurso: $[-10m, 30m]$

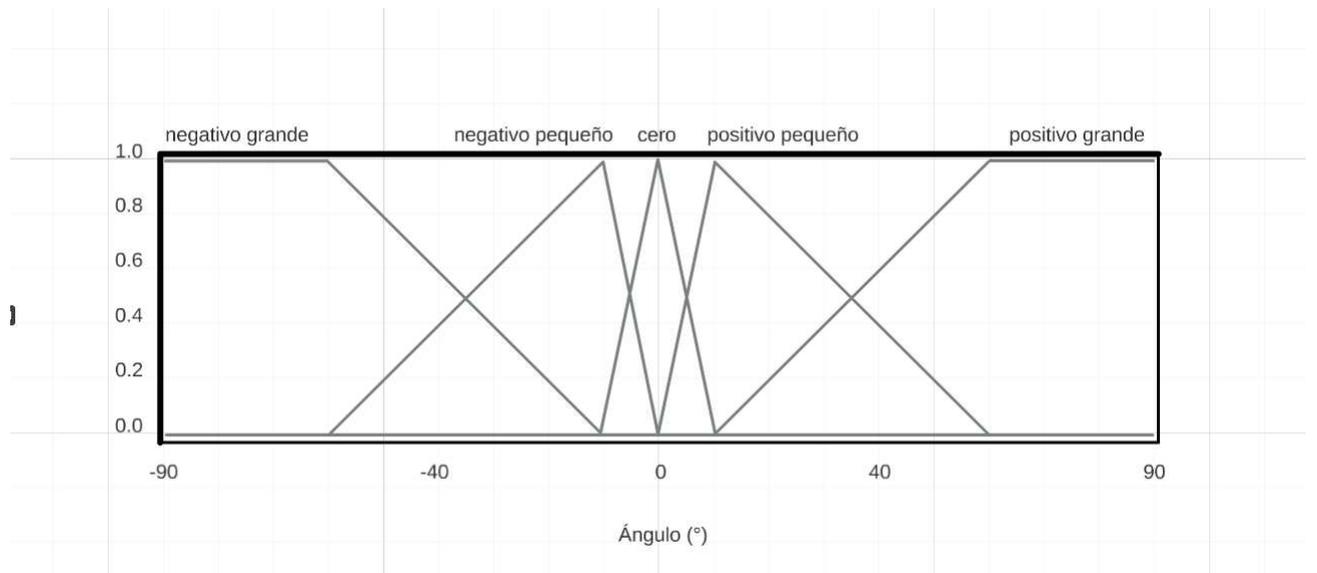
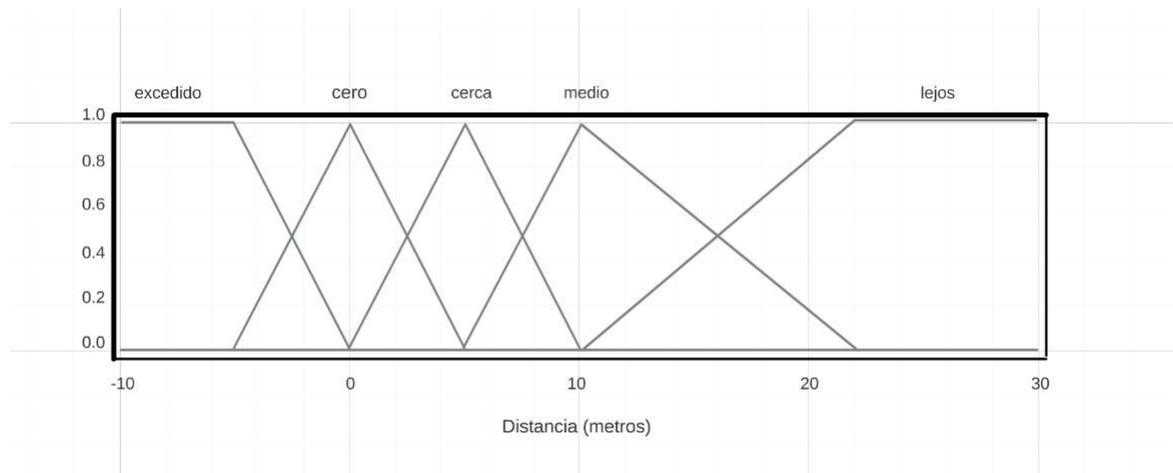
Conjuntos difusos: lejos medio, cerca, cero, excedido.

c) Potencia del motor: Universo de discurso: $[-27KW, 30KW]$

Conjuntos difusos: positivo alto, positivo medio, cero, negativo medio, negativo alto.

Base de Reglas Difusas. A partir del conocimiento obtenido empíricamente por los operadores de la grúa, los expertos en este dominio, se ha definido la siguiente base de reglas difusas:

R_1 : SI el ángulo es positivo pequeño y la distancia es cero ENTONCES la potencia es negativa media.



R_2 : SI el ángulo es cero y la distancia es cero ENTONCES la potencia es cero.

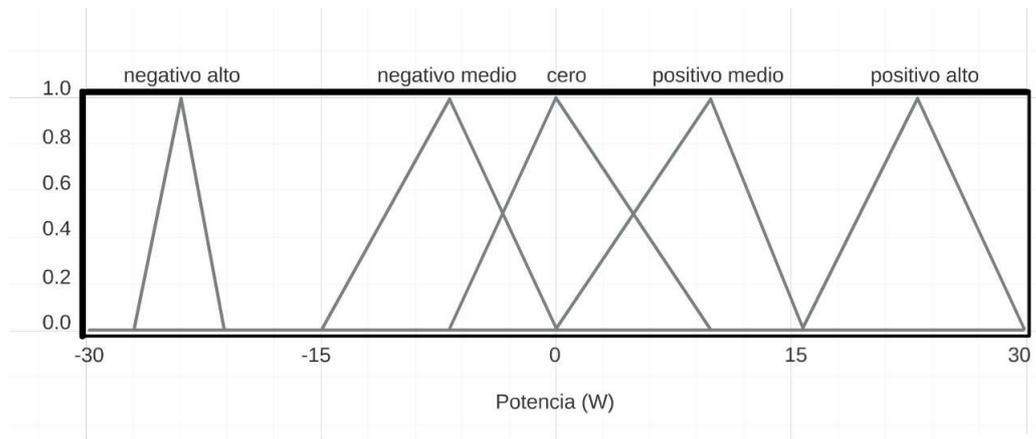
R_3 : SI el ángulo es positivo pequeño y la distancia es cerca ENTONCES la potencia es negativa media.

R_4 : SI el ángulo es cero y la distancia es cerca ENTONCES la potencia es cero.

R_5 : SI el ángulo es negativo pequeño y la distancia es cerca ENTONCES la potencia es positiva media.

R_6 : SI el ángulo es negativo pequeño y la distancia es media ENTONCES la potencia es positiva alta.

R_7 : SI el ángulo es negativo grande y la distancia es media ENTONCES la potencia es positiva media.



R_8 : SI el ángulo es cero y la distancia es lejos ENTONCES la potencia es positiva media.

R_9 : SI el ángulo es negativo pequeño y la distancia es lejos ENTONCES la potencia es positiva alta.

Fusificación. De acuerdo a las gráficas de las funciones de pertenencia, cuando $\text{ángulo}=40^\circ$ las funciones de pertenencia activadas son:

$$f_{\text{positivo pequeño}}(40) = 0.4, f_{\text{positivo grande}}(40) = 0.6$$

y para distancia= 2m son:

$$f_{\text{cerca}}(2) = 0.4, f_{\text{ceros}}(2) = 0.6$$

Inferencia. A partir de la fusificación de los datos de entrada se observa que las reglas activadas son:

R_1 : SI el ángulo es positivo pequeño y la distancia es cero ENTONCES la potencia es negativa media.

R_3 : SI el ángulo es positivo pequeño y la distancia es cerca ENTONCES la potencia es negativa media.

Con estas reglas hacemos los cálculos:

a) Cálculo de la parte SI:

Debido a la estructura de las reglas difusas activadas, hacemos el cálculo de la parte SI usando el operador Y:

$$R_1: \min(0,4, 0,4) = 0,4 = \alpha_1$$

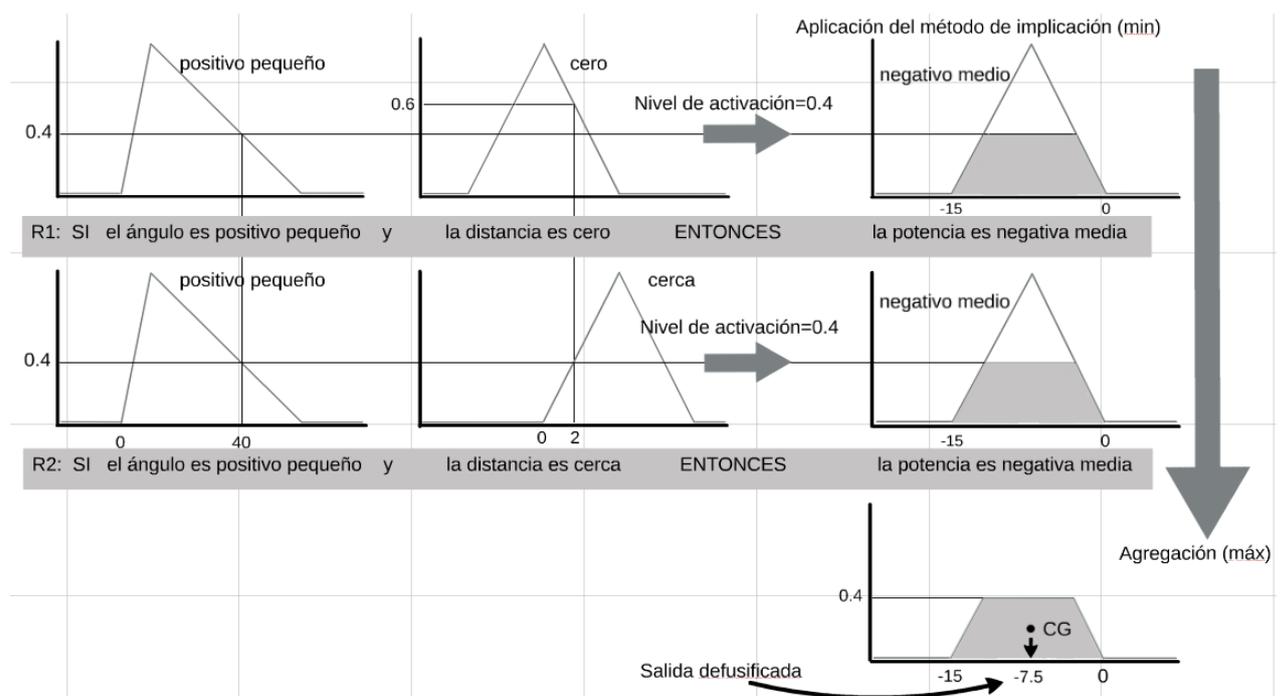
$$R_3: \min(0,6, 0,4) = 0,4 = \alpha_2$$

b) Cálculo de la parte ENTONCES:

En esta parte usamos implicación Mamdani, al truncar la gráfica del conjunto difuso consecuente de cada regla activada con el nivel de activación correspondiente α_j calculado en la parte SI, que para este problema en particular es igual a 0,4 en cada regla activada.

c) Agregación:

El conjunto difuso de salida será entonces la agregación difusa de los conjuntos difusos conclusión calculados en la parte entonces. La gráfica se observa en la figura:



Defusificación. Usando el Método de defusificación del Centro de Gravedad en el que se aplica la fórmula (1.25), se observa que el resultado a tomar es la abscisa del centro de gravedad geométrica de la figura formada entre el gráfico del conjunto difuso obtenido en la agregación y el eje de abscisas, que en nuestro caso es un trapecio. Por la simetría de la figura, este valor es el punto medio entre los extremos de la base del trapecio, es decir igual a $-7,5KW$.

Terminamos esta sección, resaltando nuevamente la utilidad de usar un sistema de inferencia difuso para modelar de manera aproximada un proceso que resulta muy difícil de describir matemáticamente. Suele pasar en los sistemas donde la dependencia entre las variables de estudio es demasiado compleja o caótica como para ser modelada mediante alguna ecuación.

Capítulo 2

Fundamentos Hidrológicos Generales

2.1. Nociones Básicas de Hidrología

El Agua en la Tierra

El agua se encuentra en la Tierra en cantidades considerables en sus tres estados físicos: líquido, sólido y gaseoso. Así también en la atmósfera, mares, océanos y los continentes. El agua pasa fácilmente de una fase a otra y de un ambiente a otro, a este comportamiento dinámico según el tiempo y espacio se le denomina **ciclo hidrológico** o **ciclo del agua**.

El Ciclo Hidrológico

Se asume que por acción del sol y de la fuerza de gravedad, el ciclo hidrológico o ciclo de agua se inicia con la evaporación de las superficies libres de agua, este vapor resultante es transportado por el viento que al llegar a cierta altura se condensa para formar las nubes, éstas darán origen a las precipitaciones bajo sus diferentes formas: líquida o sólida.

Parte de esta precipitación se evapora rápidamente en la atmósfera. Sin embargo, la mayor parte llega hasta la superficie de la tierra en donde ocurre la evaporación desde el suelo o desde la hoja y tallos de las plantas sobre las que ha caído (transpiración), otra se infiltra ingresando en el suelo a varios niveles en la que reaparecerán bajo la forma de manantiales (**escorrentía subsuperficial**) o constituirá las napas de agua subterránea (**escorrentía subterránea**).

Del agua precipitada y que no se ha infiltrado o evaporado (**escorrentía superficial**), se forman los cursos de agua superficiales como riachuelos, ríos, los que van a desembocar en lagos y mares y océanos desde donde comienza nuevamente el ciclo.

Volumen de Agua del Ciclo Hidrológico

Según datos estimativos publicados por la UNESCO en 1978, el volumen total de agua que participa en el ciclo hidrológico es de 1386 millones de kilómetros cúbicos aproximadamente, de los cuales:

- El 97,5 %, es agua salada.
- El 2,24 % es agua dulce está conformada por las aguas congeladas en las profundidades de la Antártida y en las aguas subterráneas profundas.
- El 0,26 % es agua dulce accesible para el consumo y se encuentra en los lagos, embalses, y suelos de los acuíferos poco profundos.

Esta mínima fracción de agua accesible son los principales componentes de los recursos hídricos en la Tierra y dependen directamente de la precipitación y por el deshielo de los glaciares de algunas zonas, y completados por el rocío y el goteo de niebla en ciertos lugares, es decir es la única fuente disponible del ciclo hidrológico en régimen sostenible.

Cuencas y Vertientes

Entendemos por **cuenca hidrográfica** a toda el área o superficie del terreno que aporta sus aguas de escorrentía a un mismo punto de desagüe o punto de cierre. Una cuenca está formada por un entramado de ríos, arroyos y/o barrancos de mayor o menor entidad que conducen los flujos de agua hacia un cauce principal, que es el que normalmente da su nombre a la cuenca; su perímetro es una línea curvada y ondulada que recorre la divisoria de vertido de aguas entre las cuencas adyacentes.

Por otro lado, una **vertiente hidrográfica** es un conjunto de cuencas hidrográficas cuyos ríos con sus afluentes desembocan en un mismo mar y en ocasiones, en un mismo lago, especialmente, si es de superficie considerable. Es un concepto cuya utilidad se debe a que integra ríos con características similares y cuyo estudio, por lo tanto, puede tener ciertas aplicaciones interesantes en el análisis espacial tanto de países individualmente considerados como de continentes, sobre todo, en lo que se refiere al inventario de recursos.

El Agua en el Perú

El Perú posee una superficie de 1 285 200 km² con una población de cerca de 31 millones de habitantes. Se encuentra localizado en la parte central y occidental de América del Sur. Hidrográficamente el territorio peruano se divide en tres vertientes: la **Vertiente**

AMAZONAS				PACÍFICO			
RÍOS	LONGITUD ESTIMADA (Km.)	RÍOS	LONGITUD ESTIMADA (Km.)	RÍOS	LONGITUD ESTIMADA (Km.)	RÍOS	LONGITUD ESTIMADA (Km.)
Ucayali	1 771	Trapiche	448	Camaná	375	Grande	169
Marañón	1 414	Inambari	437	Chira	334	Sama	163
Putumayo	1 380	Curaray	414	Santa	316	Huaura	159
Yavarí	1 184	Morona	402	Tambo	287	Chicama	157
Huallaga	1 138	Tambopata	402	Piura	267	Jequetepeque	153
Urubamba	862	Pachitea	393	Ocoña	262	Rímac	146
Mantaro	724	Majes (Camaná)	388	Chili	240	San Juan	140
Amazonas	713	Aguayfia	379	Ica	209	Caravelí	140
Apurímac	690	Pampas	379	Cañete	206	Moquegua	131
Napo	667	Nanay	368	Acarí	195	Motupe	124
Madre de Dios	655	Pastaza	368	Locumba	187	Chillón	124
Piedras	621	Manu	356	Chancay	183	Mala	122
Tigre	598	Tamaya	310	Pisco	175	Chancay	117
Purús	483	Carabaya (Azángaro)	304	Yauca	172	Olmos	115
Corrientes	448	Mayo	299	Pativilca	171	Caplina	115

FUENTE: INEI, IGN

Figura 2.1: Principales Ríos del Perú

del Océano Pacífico con un área de 283 600 km² (22 %), **la Vertiente del Amazonas** con 952 800 km²(74 %), y **la Vertiente del Lago Titicaca** con 48 800 km²(4 %).

La Vertiente del Amazonas es la más húmeda, presenta precipitaciones que fluctúan entre 1500 mm/año y 4000 mm/año, y su escorrentía media anual fluctúa entre 300 mm/año y 500 mm/año. La masa anual promedio de agua superficial que producen las 106 cuencas de las vertientes del territorio peruano es de 780 000 MMC (millones de metros cúbicos), pero el 90 % es agua que se va al Atlántico a través del río Amazonas, y del otro 10 % sólo se aprovecha una pequeña parte, debido al régimen estacional de nuestras corrientes de agua. (Fuente: SENAMHI).

2.2. Aspectos Generales de los Procedimientos Hidrométricos.

La **Hidrometría** es la parte de la hidrología que se encarga de tomar y proveer datos relacionados con la distribución espacial y temporal del agua sobre la tierra. Esta información es un insumo fundamental para proyectos de planeamiento y manejo de los recursos hídricos, para los cuales es de vital importancia conocer las variaciones hidráulicas (caudales, niveles, velocidades, etc.) de cada una de las corrientes y cuerpos de agua.

Se definen los siguientes conceptos:

1. **Caudal.** Volumen de agua que pasa a través de una sección transversal de una corriente en la unidad de tiempo; generalmente se expresa en metros cúbicos por segundo (m³/s) o litros por segundo (l/s).
2. **Nivel.** Elevación de la superficie del agua por encima o por debajo de algún indicador, el cual debe estar sujeto a un sistema de referencia fijo preestablecido. Hay varias formas de medir el nivel, incluyendo el uso de escalas hidrométricas o limnímetros (más utilizados en la actualidad), cadenas, cintas, etc.
3. **Aforo Líquido.** Es un procedimiento que consiste en realizar una serie de mediciones en campo que permiten posteriormente calcular el caudal de una corriente. Este caudal debe estar asociado a un nivel del agua.

El caudal puede medirse por diferentes métodos. La elección del método de aforo depende de las condiciones específicas de cada sitio; la mayoría de estos métodos

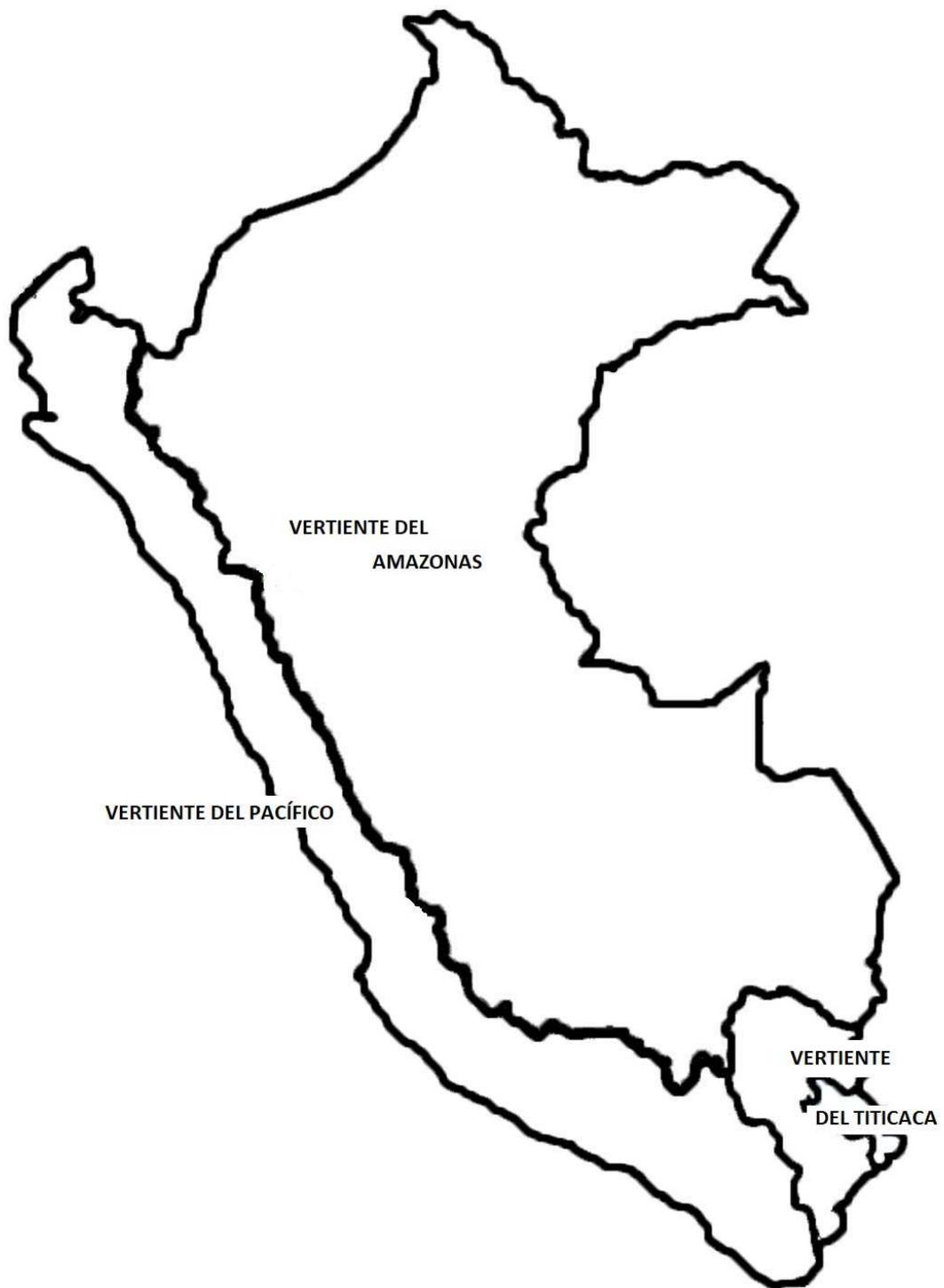


Figura 2.2: Vertientes Hidrográficas del Perú

están basados en la medición de la velocidad y el área en una sección transversal determinada.

Generalmente se efectúan aforos periódicos para determinar la relación entre el nivel del agua y el caudal en la estación hidrométrica, con estos datos se construye la curva de gastos o curva de calibración nivel de agua-caudal con la cual es posible determinar el flujo cuando sólo se cuenta con el dato de nivel.

4. **Sección transversal.** Corte producido en un punto de un cauce por medio de un plano vertical perpendicular a la dirección del flujo. El establecimiento de secciones transversales en las corrientes se emplea para la realización de aforos líquidos, medición de parámetros de calidad de agua, medida de niveles, etc.

2.2.1. Requerimientos para la selección de la sección de aforo.

El sitio seleccionado para efectuar los aforos deberá cumplir, tanto como sea posible, con los siguientes requerimientos:

- a) El canal en el sitio de medición debe ser recto y tener sección transversal uniforme y una pendiente que minimice las distribuciones anormales de velocidad.
Nota: cuando la longitud del tramo recto del río es limitada, para hacer las mediciones con molinetes u otros métodos de determinación de velocidad se debe ubicar la sección de tal manera que la longitud aguas arriba sea por lo menos dos veces la de aguas abajo.
- b) Las direcciones de flujo para todos los puntos en cualquier vertical a través del ancho del cauce deben ser paralelas unas a otras y perpendiculares a la sección transversal. Los sitios que presenten vórtices, flujo reversible o zonas muertas deben ser rechazados.
- c) El lecho y las márgenes del río deben ser estables y bien definidas para todas las condiciones del flujo, de tal manera que se pueda hacer una medición precisa del área de la sección transversal.
- d) Las curvas de distribución de velocidades deben ser regulares en los planos verticales y horizontales de medición.
- e) Las condiciones de la sección y de su entorno no deben ocasionar cambios en la distribución de la velocidad durante la medición.

- f) La sección de medición debe ser claramente visible a través del ancho y no debe presentar obstrucciones como árboles, plantas acuáticas u otros. Cuando se afora desde un puente con varios estribos que dividen la sección transversal cada subsección del canal debe ser tratada individualmente.
- g) La profundidad del agua en la sección debe ser suficiente para proveer una inmersión efectiva del molinete en cualquier punto en que se mida.
- h) El sitio debe tener fácil acceso en todo momento y contar con los equipos necesarios de medición.
- i) La sección debe estar ubicada lejos de estaciones de bombeo y vertimientos. Si éstos operan durante las mediciones, probablemente crearán condiciones inconsistentes con la relación natural Nivel-Caudal que se tiene para la estación.
- j) Los sitios donde hay convergencia o divergencia de flujo deben descartarse.
- k) En donde sea necesario hacer mediciones en cercanías de un puente, es preferible que el sitio de aforo esté ubicado aguas arriba del mismo.
- l) En ciertas ocasiones es necesario verificar datos haciendo aforos en secciones diferentes a la seleccionada para la estación. Esto se puede hacer siempre y cuando no haya cambios sustanciales por pérdidas o ganancias de caudal ocasionados por intervenciones y cuando las mediciones se relacionen con los niveles registrados en la sección principal.

Nota: si el sitio ha sido estabilizado para una estación permanente o para ser usado en futuras mediciones, debe estar provisto de medios para demarcación de la sección transversal y para determinación de los niveles, de tal forma que sea fácilmente identificable en otras mediciones y los datos tomados sean comparables con datos anteriores.

2.2.2. Demarcación de la sección de aforo para el estudio del problema.

Cuando se ha seleccionado una sección idónea para el aforo y sobre ésta se inicie un programa de medición periódico, bien sea permanente o transitorio, la sección se debe demarcar así:

- a) Debe definirse en las dos bancas con marcas claramente visibles y fácilmente identificables la posición de la sección transversal, normal a la dirección de flujo.
- b) Los niveles durante la medición deben leerse en una mira (limnómetro), cuyo dátum o cero debe estar asociado a un sistema de referencia común a nivel nacional o regional
- c) Cuando se considere que existen diferencias de nivel de agua que sean importantes entre las dos bancas del río, debe instalarse una referencia auxiliar en la banca opuesta. Esto toma importancia particular en el caso de ríos muy anchos.

2.2.3. Estaciones Hidrométricas

Una estación hidrométrica es un lugar ubicado en una sección transversal del río donde se colocan instrumentos de medición con los cuales se obtienen datos para realizar un conjunto de operaciones que permiten determinar el comportamiento de las variables de interés en estudio, tales como el nivel y el caudal en un tiempo determinado. Las estaciones hidrométricas se pueden clasificar según:

a) El tipo de mediciones hidrométricas.

- **Registro de nivel de agua:** Son aquellas estaciones en las cuales sólo se toman datos de nivel de agua y no se realizan aforos, por lo tanto no es posible definir una curva de calibración Nivel-Caudal. La frecuencia de medición en ellas depende de las necesidades de información. -
- **Registro de nivel y caudal:** En este tipo de estaciones, adicional a la frecuencia de lecturas diarias de nivel, se realizan mediciones de las descargas asociadas a su respectivo nivel de agua instantáneo. Con estos pares de datos se va construyendo y/o actualizando la relación Nivel-Caudal para la estación.

b) El tipo de instrumentación.

- **Manuales:** Son estaciones en las cuales la medición o lectura la realiza de forma directa la persona encargada, mediante el respectivo registro en los formatos destinados para ello; la frecuencia de medición está preestablecida, bien sea para datos diarios, horarios, así como, para intervalos de tiempo

en el día.

- **Automáticas:** Son estaciones que cuentan con equipos de registro continuo como limnógrafos. Estos son instrumentos que permiten obtener datos del nivel del agua con una frecuencia alta. Este tipo de estaciones se clasifican en dos grupos:
 - . **Con registro análogo:** en donde los mecanismos que se encargan de transmitir el nivel del agua desde el pozo de quietamiento al registrador lo hacen de forma gráfica, es decir, se imprimen en una banda de papel.
 - . **Con registro digital:** En donde, además de contar con el registro análogo, se tienen equipos que convierten el nivel transmitido por los mecanismos en una señal digital que puede ser almacenada y en algunos casos transmitida a un centro de control lejos de la estación. Estas se denominan estaciones telemétricas.

c) Frecuencia de Operación.

- **Anuales:** Son aquellas estaciones que por su difícil acceso no permiten una frecuencia mayor de operación; por lo tanto, deben tener equipos especiales que permitan hacer un registro continuo durante un período hidrológico largo.
- **Estacionales:** Son aquellas en las que su operación tiene en cuenta la recolección de información para un período específico de tiempo, en el cual se considera importante obtener los datos; estos períodos pueden asociarse al registro de caudales extremos (máximos o mínimos).
- **Mixtas:** Son estaciones en las cuales su operación se hace de manera continua y con una frecuencia específica en la cual se pueden medir datos extremos, bien sea máximos o mínimos, de manera aleatoria y en donde la mayoría de las mediciones corresponde a situaciones promedias.

2.2.4. Equipos de medición

Los parámetros básicos a determinar cuando se ejecuta un aforo líquido (exceptuando los métodos de aforo por trazadores y volumétrico) son la profundidad y la velocidad

de flujo en las verticales, el nivel del agua y el ancho de la sección transversal. La medición de estas variables se lleva a cabo mediante instrumentos apropiados, los cuales se seleccionan según el método de aforo empleado. A continuación se describen este tipo de instrumentos:

a) Medidores de la Velocidad del Agua.

La medición de la velocidad del agua (para los métodos de vadeo y suspensión) se hace utilizando un equipo denominado molinete o correntómetro (Figura 2.3 ¹). Los molinetes más empleados son el de cazoletas (con eje vertical) y el de hélice (de eje horizontal). Las revoluciones efectuadas por la hélice del molinete se registran en un contador. Los molinetes se calibran para un rango de velocidades dado. La relación entre la velocidad del flujo y la velocidad del rotor se expresa, por lo general, en revoluciones por segundo (rev/s).



Figura 2.3: Correntómetro

b) Contadores.



Figura 2.4: Contador

¹Fuente: http://ayudahispano-3000.blogspot.pe/2015/09/instrumentos-de-medicion_28.html

Son equipos que se encargan de registrar el número de revoluciones dadas por la hélice de un molinete en un tiempo determinado cuando se va a calcular velocidad de flujo en una corriente. El contador por cada giro de la hélice recibe un impulso eléctrico y lo convierte en una señal análoga o digital, permitiendo así establecer el número de revoluciones (Figura 2.4 ²).

c) Limnímetros.

El nivel del agua en una estación se registra por medio de una mira o limnómetro (Figura 2.5³), el cual consiste en una regla graduada que permite determinar la observación de las fluctuaciones de los niveles del cuerpo de agua. Se ubica en un punto fijo sobre la sección transversal y permanece en contacto con la corriente. Cuando el talud de la sección transversal es muy tendido (de pendiente muy baja) y el nivel del agua en el cauce puede presentar grandes variaciones se suele instalar una serie de limnímetros, cada uno de los cuales permitirán registrar diferentes rangos de variación de esta variable.



Figura 2.5: Limnómetro

d) Limnógrafos.

²Fuente: <https://www.aliexpress.com/item/Free-Shipping-Waterproof-Marine-Tachometer-RPM-Counter-Snowmobile-Skis-Motor-Bike-Go-Kart/32231860721.html>

³Fuente: <http://www.engenhariacivil.tecnilab.pt/conteudo/areas-actuacao/engenhariacivil/lecg-sist-monitorizacao-symest/lecg-sist-monitorizacao-symest-equipamentos/limnometro>

En lugares en donde el nivel de agua cambia con mucha frecuencia es necesario instalar equipos especiales de registro continuo llamados limnógrafos (Figura 2.6⁴) por medio de los cuales es posible obtener datos del nivel del agua en función del tiempo con una alta frecuencia, los cuales son representados en un limnigrama (Figura 2.7⁵). Estos equipos están compuestos fundamentalmente por 3 dispositivos: el primero corresponde al elemento sensible (flotador y contrapeso o manómetro), el segundo es el sistema que traduce a escala y registra los niveles del agua (eje helicoidal, poleas de escala y sistema inscriptor y de registro), y el tercero, basado en un mecanismo de relojería, alimentado mecánicamente (cuerda) o por medio de baterías (pilas de 6V), que proporciona una escala de tiempo.



Figura 2.6: Limnógrafo

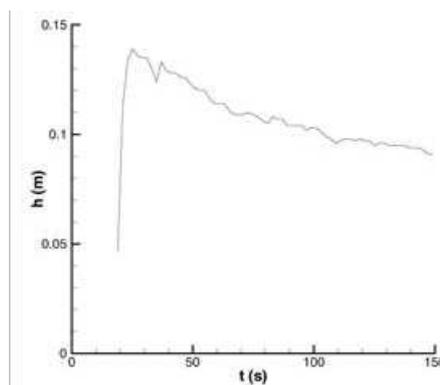


Figura 2.7: Limnigrama

e) Malacates.

⁴Fuente: <http://www.dicyt.com/viewNews.php?newsId=7515>

⁵Fuente: http://hispagua.cedex.es/sites/default/files/hispagua_articulo/Ingcivil/2001/121/articulo2/articulo.html



Figura 2.8: Malacate

Para la manipulación del conjunto molinete-lastre en los aforos que se realizan por suspensión es necesario contar con un equipo (malacate) que facilite su manejo teniendo en cuenta que éste normalmente es pesado (entre 30 y 75 lb.) (Figura 2.8⁶). Los malacates, además de permitir manipular grandes pesos con relativa facilidad, están provistos de mecanismos especiales (contador) para registrar la longitud desplegada del cable y de esta manera obtener la medición de la altura de suspensión y la profundidad de la corriente.

- f) Equipos para Medición de la sección transversal.-** Se considera el ancho de la sección húmeda y se mide utilizando una cinta métrica convencional (cuando las condiciones así lo permiten) o equipos topográficos. En estaciones de medición preestablecidas, el ancho se determina mediante el abscisado existente, el cual se encuentra referenciado o marcado sobre la estructura (bien sea puente o tarabita⁷)⁸.

2.3. El río Rímac

El río Rímac, el más importante del departamento de Lima, nace en las alturas de Ticlio discurriendo a través de las provincias de Lima y Huarochirí con dirección noreste - suroeste y una longitud de 140 Km. Sus afluentes principales son la quebrada

⁶Fuente: https://www.engormix.com/proyeccion-tecnica-agropecuaria/malacates-dutton-laison-sh11802_pr22253.htm

⁷Tarabita: Cuerda gruesa a través de la cual se desplaza una silla o caja grande que sirve para transportar personas o carga desde una orilla de un río a la otra

⁸Fuente de la imagen 2.9: <https://www.picquery.com/c/tarabita-mindo>



Figura 2.9: Tarabita

El Carmen y, los ríos Chinchán, Blanco, Aruri, Santa Eulalia y Huaycoloro. Los centros poblados más importantes ubicados a lo largo del río hablador son: Lima, Vitarte, Chaclacayo, Chosica y Matucana que representan el 81 % de la población total de la cuenca. Es importante señalar que la capital de la República es la principal consumidora del agua superficial y acuífero del río Rímac⁹.

El río Rímac es un río de régimen regular (caudal permanente), cuyo caudal es sometido a un uso intensivo a lo largo de su trayectoria. En la cuenca alta, el agua es utilizada en la actividad minera, en la cuenca media, utilizada para la generación de energía eléctrica, asimismo en ésta parte de la cuenca y la cuenca baja (desde Matucana hasta el Callao) el agua se usa en agricultura, y principalmente como fuente de abastecimiento de agua para la ciudad de Lima Metropolitana.

Sin duda, la cuenca del río Rímac es considerada una de las cuencas hidrográficas más importantes del Perú, porque es la fuente de agua superficial que asegura el abastecimiento de agua potable a aproximadamente 7 millones de personas (27 % de la población peruana) que habitan en la ciudad de Lima Metropolitana y en otras zonas de la cuenca([25]).

⁹Fuente: http://www.ana.gob.pe/sites/default/files/normatividad/files/rio-rimac-febrero-2011_0_0.pdf

Capítulo 3

Planteamiento del Problema

El agua de los ríos representa el elemento vital para la supervivencia del hombre, mas aún cuando ésta se utiliza para distintos usos, entre los de mayor importancia están: consumo poblacional, agrícola, pecuario, minero, energético y transporte. Por esta razón es importante un estudio que permita el análisis de escenarios futuros que puedan llegar a definir criterios de uso, así mismo para el planeamiento y gestión de las políticas que permitan la toma de decisiones para afrontar los eventos extremos de una forma adecuada. Así, en el año 2009, bajo la ley de Recursos Hídricos (Ley N° 29338) se creó el Sistema Nacional de Gestión de los Recursos Hídricos (SNGRH) el cual está conformado por un conjunto de instituciones gubernamentales y privadas, que establecen principios, normas, procedimientos, proyectos, técnicas e instrumentos mediante los cuales el Estado desarrolla, asegura y promueve la gestión integrada, participativa multi-sectorial, el aprovechamiento sostenible, la conservación, preservación de la calidad y el incremento de la disponibilidad de los recursos hídricos. Bajo esta misma ley se creó la Autoridad Nacional del Agua (ANA) en el año 2008 y cuyas principales funciones son : Ejercer jurisdicción administrativa exclusiva en materia de aguas, desarrollando acciones de administración, fiscalización, control y vigilancia, para asegurar la conservación y protección del agua en cuanto a su cantidad, además promover y apoyar la ejecución de proyectos que incentiven el uso eficiente, el ahorro, la conservación, la protección de la calidad e incremento de la disponibilidad de los recursos hídricos.

La labor de realizar estudios para evaluaciones y predicciones respecto a este recurso hídrico en el Perú la realiza el Servicio Nacional de Meteorología e Hidrología (SENAMHI) a través de sus publicaciones online y/o impresa las mismas que sintetizan

el comportamiento hidrológico de las principales cuencas del territorio peruano registradas permanentemente y sus comparaciones en relación al historial hídrico de niveles hidrométricos y caudales. Mensualmente SEHAMHI edita un boletín hidrológico que proporciona información sobre el comportamiento de las variables antes mencionadas, a escala diaria y mensual, condición que permite realizar el monitoreo, evaluación y análisis de la información, permitiendo posteriormente la publicación para conocimiento de las diferentes instituciones y público en general, a fin de contribuir en la Planificación y Gestión del Recurso Hídrico , pues el país enfrenta un gran desafío en materia de optimización del aprovechamiento y conservación del recurso.

3.1. Técnicas usadas actualmente por SENAMHI.

El Centro de Predicción Numérica del SENAMHI (CPN) trabaja actualmente con técnicas a corto y largo plazo que a continuación se detallan¹

Hidrological Forecasting System (HFS). Es un modelo matemático hidrológico de pronóstico a corto plazo, el cual transforma precipitaciones en caudales. Este modelo fue desarrollado por el Centro de Investigación Hidrológica del Estado de California (HRC) en conjunto con el Servicio Meteorológico Nacional de Estados Unidos, el cual está operativo desde finales del 2000. En la actualidad genera caudales diarios y máximos instantáneos pronosticados con una anticipación de hasta 5 días para la cuenca del río Rímac, esperándose implementar en el futuro el sistema para otras cuencas importantes del país. Los datos de salida son los caudales simulados y pronosticados así como la incertidumbre del caudal simulado.

Sacramento EHF. Fue instalado y configurado con asesoría de la Dra. Carpenter (HRC), este es un modelo de pronóstico a largo plazo, el cual utiliza el Modelo Sacramento HFS con una metodología de pronóstico ensamblado. Actualmente, genera salidas para cuatro meses de validez de los caudales diarios del río Rímac y está implementándose también para el río Piura. Utiliza de entrada datos históricos de precipitación, evapotranspiración potencial, parámetros del modelo y estados del modelo al comienzo del periodo del pronóstico.

HEC-RAS. A diferencia de los modelos anteriores, este modelo no predice caudales, por el contrario, necesita ser provisto de estos datos y trabaja con la geo-

¹Fuentes: <http://www.senamhi.gob.pe/site/ois/presentacion.pdf>; <http://www.senamhi.gob.pe/?p=0602>

grafía del tramo del río en estudio para realizar simulaciones de áreas inundables, detección de zonas de desborde, etc. Esta información servirá para la efectiva administración de recursos hídricos así como para mitigar los efectos de desastres naturales en zonas vulnerables.

Los modelos HFS y EFS son modelos de lluvia-escorrentía, las predicciones a partir de este tipo de modelos pueden ser en ocasiones insatisfactorias debido a que la variabilidad espacial de la lluvia no esté probablemente bien representada en regiones donde los datos son escasos. Para un mejor detalle, es necesario el monitoreo de grandes extensiones de territorio. La incertidumbre de la estimación de la precipitación se incrementa con la disminución de la densidad de estaciones pluviométricas, tal es el caso de los Andes y Amazonía peruana, caracterizado por una red de estaciones pobremente distribuida.

3.2. La propuesta de trabajo

Ante la problemática descrita anteriormente, surge la necesidad de construir un modelo de predicción de niveles y caudales en el que los insumos (entradas del modelo) sean más fácilmente obtenidos y no dependan tanto de un amplio monitoreo de la zona, pero que a la vez, por prescindir de estas características, no genere mucha incertidumbre en las predicciones resultantes. Entonces podemos empezar haciendo algunas suposiciones que nos permitirán concebir de manera más simple el modelo a seguir:

1. Existe una función continua que ajuste de manera óptima los registros históricos y describa el comportamiento (al menos a corto plazo) de los datos de niveles y caudales.
2. La función del ítem anterior podría ser tan compleja de describir que su funcionamiento podría ser aproximado por medio de un sistema de inferencia difuso.
3. Puesto que es innegable que existen otros factores (precipitaciones, cambio estacional, geomorfología de la zona, etc) que determinan el nivel y caudal resultante a futuro y es imposible medirlos, podemos considerar que estos parámetros serán incluidos implícitamente en nuestro modelo si hacemos que la función en cuestión dependa de algunos valores pasados de niveles y caudales.

De acuerdo a lo que se ha visto en el primer capítulo, una red neuronal es una buena herramienta para aproximar una función continua. Por otro lado, para incorporar la dependencia de valores anteriores de niveles y caudales dentro del modelo, tendremos que trabajar con una red neuronal recurrente. El aporte de la lógica difusa a esta red será mostrado en el siguiente capítulo.

Capítulo 4

Una nueva metodología: Red Neuro-Difusa Recurrente.

Como se ha visto en el primer capítulo, una red neuronal es un buen aproximador de una función continua, por eso es natural pensar en usarla para describir la relación entre un conjunto de variables que siguen un comportamiento continuo en el tiempo. Sin embargo, la complicación de usar este tipo de sistemas puede surgir al momento de determinar el número de capas ocultas a usar. Por otro lado, la tendencia altamente no lineal y caótica de la relación entre las variables de estudio de nuestro caso (caudales y niveles) sugiere la intervención de la lógica difusa en nuestro modelo. Es así como surge la idea de una red neuronal difusa. En ésta, reglas SI-ENTONCES son incorporadas en las entradas, dándonos la oportunidad de modificar la cantidad de reglas (y por lo tanto el número de funciones de pertenencia) en lugar de el número de capas de la red.

4.1. Un modelo neuronal difuso

La estructura presentada aquí ha sido propuesta por primera vez por Yamakawa en ([14]) y es llamada red neuronal difusa. El concepto asociado con esta estructura modifica el modelo convencional de neurona que posee pesos sinápticos constantes seguido de una función de activación (usualmente no lineal). Una neurona difusa posee sinapsis no lineales caracterizadas por reglas SI- ENTONCES con el valor de un peso sináptico como consecuente. La figura 4.1 muestra la representación gráfica de una red neuronal difusa.

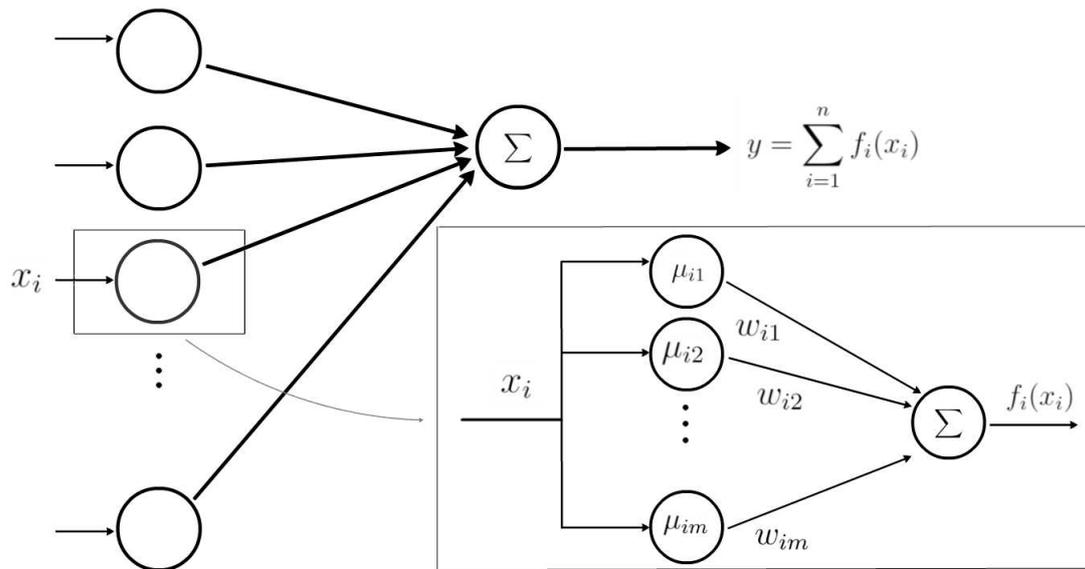


Figura 4.1: Red Neuronal Difusa

Aquí, x_i representa a la señal de entrada de la i -ésima neurona de la capa de entrada. La función de activación de la i -ésima neurona de la capa de entrada es denotada por f_i . Para construir las funciones f_i identificamos primero los valores máximo $x_{\text{máx}}$ y mínimo $x_{\text{mín}}$ que pueden tomar las entradas x_i , luego el **espacio de entrada** para esta variable se define como el intervalo $[x_{\text{mín}}, x_{\text{máx}}]$, el cual es dividido en varios segmentos difusos (funciones de pertenencia triangular) que son caracterizados por las funciones (de pertenencia) $\mu_{i1}, \mu_{i2}, \dots, \mu_{im}$ como se muestra en la figura 4.2.

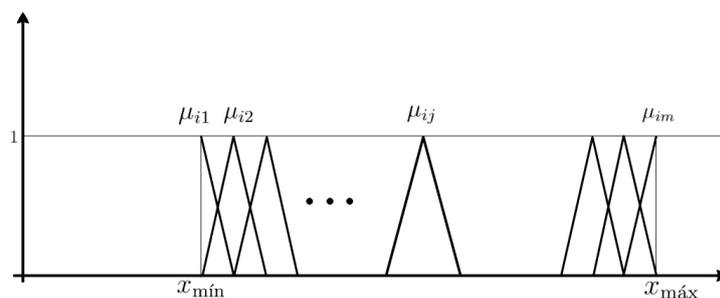


Figura 4.2: Funciones de pertenencia

Los pesos sinápticos w_{ij} asociados al j -ésimo segmento difuso de la i -ésima neurona de entrada son obtenidos por medio de la regla SI-ENTONCES siguiente:

Si la señal de entrada x_i es incluida en el segmento difuso μ_{ik} (o sea $\mu_{ik}(x_i) \neq 0$)

entonces la salida sináptica es w_{ik} .

(4.1)

Debido a que las funciones de pertenencia son complementarias (la suma de funciones de pertenencia vecinas es siempre la unidad), una señal de entrada x activa solamente una o dos reglas simultáneamente para cada sinapsis (o sea existe a lo más un valor de k para el cual $\mu_{ik}(x) \neq 0$ o $\mu_{ik+1}(x) \neq 0$).

El valor determinístico de la salida sináptica se obtiene por defusificación aplicando el *método del centro de gravedad*, dado por:

$$\begin{aligned}
 f_i(x) &= \frac{\sum_{j=1}^m \mu_{ij}(x)w_{ij}}{\sum_{j=1}^m \mu_{ij}(x)} \\
 &= \frac{\mu_{ik}(x)w_{ik} + \mu_{ik+1}(x)w_{ik+1}}{\underbrace{\mu_{ik}(x) + \mu_{ik+1}(x)}_{=1}} \\
 &= \mu_{ik}(x)w_{ik} + \mu_{ik+1}(x)w_{ik+1}.
 \end{aligned}
 \tag{4.2}$$

Así, la salida de la neurona de la capa de entrada es

$$f_i(x) = \mu_{ik}(x)w_{ik} + \mu_{ik+1}(x)w_{ik+1}. \tag{4.3}$$

Finalmente la salida y de esta red neuronal puede ser representada por la siguiente ecuación:

$$y = \sum_{i=1}^n f_i(x_i) \tag{4.4}$$

Si Y^1, Y^2, \dots, Y^N son las salidas deseadas de un conjunto de N patrones, y las salidas obtenidas por la red para el conjunto de entradas de dichos patrones son $y(1), y(2), \dots, y(N)$, entonces la función del error estará dada por

$$E = \sum_{r=1}^N (y(r) - Y^r)^2, \tag{4.5}$$

donde N es el número de patrones. Entonces, gracias a las expresiones (4.3) y (4.4), la expresión de E en (4.5) es cuadrática (parabólica) respecto a los pesos sinápticos w_{ij} .

Se ha mostrado a una red neuronal difusa como una estructura simple que usa las capacidades de una red neuronal y la lógica difusa. Como ya se ha mencionado anteriormente, la red neuronal difusa mostrada aquí fue propuesta por primera vez por Yamakawa ([14]), en donde se muestra como una gran herramienta para modelamiento de sistemas complejos a causa de su simplicidad y adaptabilidad para trabajar con sistemas no lineales ya que mientras que en una red neuronal es necesario cambiar el número de capas ocultas y el número de neuronas en cada una de ellas para mejorar la aproximación, aquí se puede cambiar el número de segmentos difusos.

4.2. Modelo de Red Neuronal Difusa Recurrente

Ahora se propone un modelo neuronal difuso modificado, el cual presenta las capacidades no lineales de la red neuronal difusa presentada por Yamakawa en ([14]). La estructura propuesta ([15]) incorpora conexiones recurrentes las cuales proveen características dinámicas. Estas propiedades hacen a la neurona un buen candidato para modelamiento de sistemas MISO (multiple input - simple output) y SISO (simple input-simple output).

4.2.1. Estructura de una Red Neurodifusa Recurrente

La estructura de una red neurodifusa recurrente es mostrada en la figura 4.3.

Las funciones de activación f_i , con $i = 1, \dots, n$ y \tilde{f}_s , con $s = 1, \dots, d$, de las neuronas que presentan conexiones hacia adelante y recurrentes respectivamente, poseen pesos sinápticos que son determinados por reglas SI-ENTONCES como las mostradas en la expresión 4.1. La salida de cada neurona de la capa de entrada es obtenida por inferencia difusa con defusificación por el método del centro de gravedad, (ver las expresiones 4.2 y 4.3), es decir:

$$f_i(x_i(r)) = \mu_{ik}(x_i(r))w_{ik} + \mu_{ik+1}(x_i(r))w_{ik+1} \quad (4.6)$$

donde los valores $x_i(r)$, con $i = 1, \dots, n$ corresponden a las entradas del sistema en el tiempo r (mejor dicho, lo que habíamos denotado por X_i^r en la expresión 1.20) y el índice k juega el mismo papel que en la expresión 4.3, es decir el único valor para el cual $\mu_{ik}(x_i(r)) + \mu_{ik+1}(x_i(r)) = 1$. Asimismo,

$$\tilde{f}_s(y(r-s)) = \tilde{\mu}_{sp}(y(r-s))\tilde{w}_{sp} + \tilde{\mu}_{sp+1}(y(r-s))\tilde{w}_{sp+1} \quad (4.7)$$

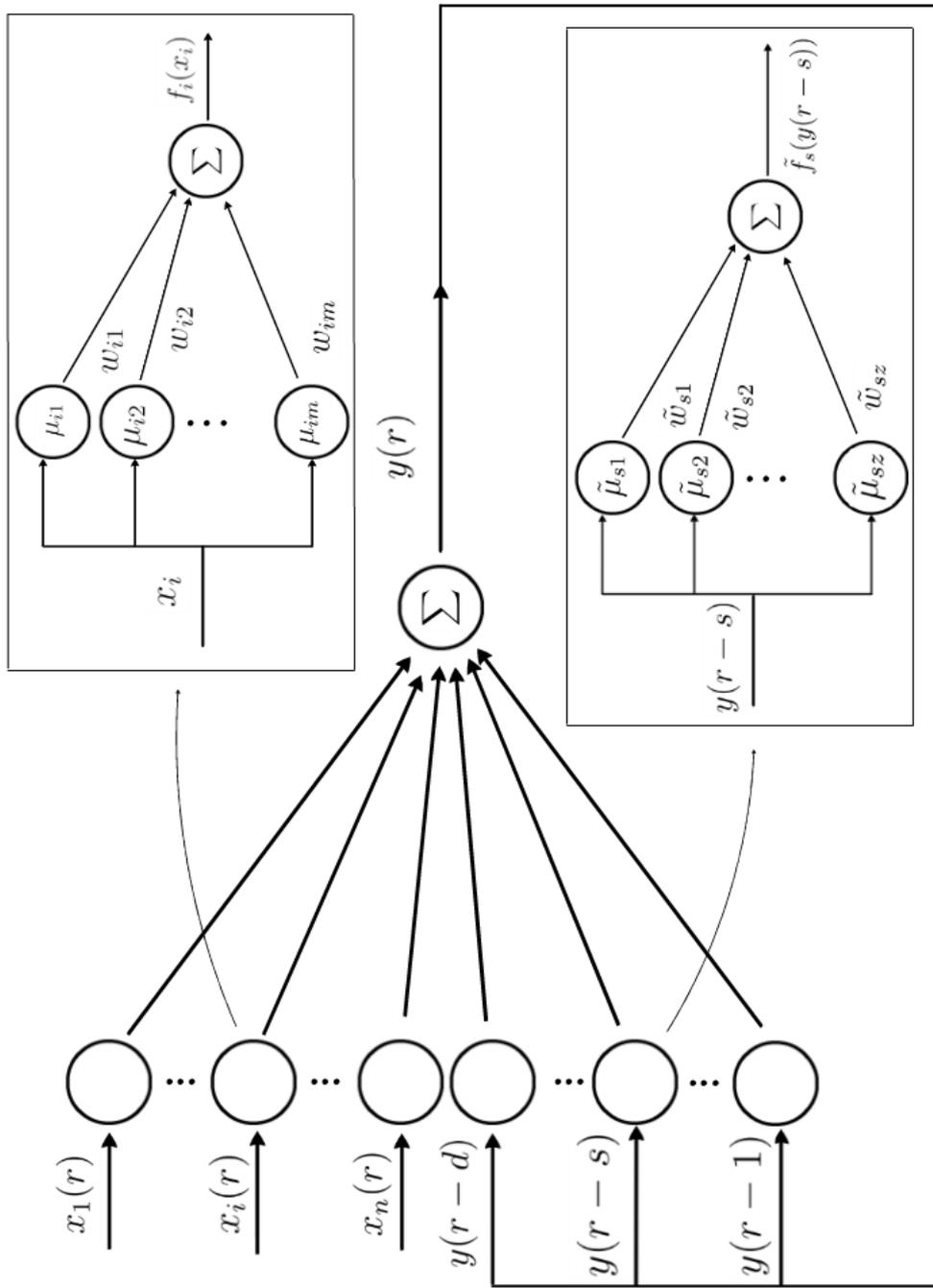


Figura 4.3: Red Neuronal Difusa recurrente

donde los valores $y(r - s)$, con $s = 1, \dots, d$ son los términos recurrentes asociados a la salida de la neurona en el tiempo $r - s$, el índice p es (el único) tal que $\tilde{\mu}_{sp}(y(r - s)) + \tilde{\mu}_{sp+1}(y(r - s)) = 1$ y d es el orden estimado de retraso del sistema.

Por lo tanto la salida de esta red neuronal difusa estará dada por

$$y(r) = \sum_{i=1}^n f_i(x_i(r)) + \sum_{s=1}^d \tilde{f}_s(y(r - s)). \quad (4.8)$$

En general, la fusificación de las variables (por medio de segmentos difusos) implica definir el número de zonas en las que se divide el espacio de entrada. Generalmente el número de regiones o zonas que caracterizan un sistema no lineal complejo no es completamente conocido, por lo tanto, es conveniente empezar el diseño de la red neuronal difusa usando un número reducido de funciones de pertenencia e ir incrementando progresivamente este número para cada variable de entrada con la finalidad de mejorar la aproximación. Un número reducido de funciones de pertenencia puede afectar la aproximación pero muchas funciones de pertenencia pueden hacer difícil el proceso computacional con propósitos de implementación.

4.2.2. Algoritmo de aprendizaje de una Red Neurodifusa Recurrente.

El algoritmo de aprendizaje general es obtenido en términos del método de la pendiente máxima. Supongamos que disponemos de un conjunto de N patrones de entrada-salida, entonces la función de error E se define como:

$$E = \frac{1}{2N} \sum_{r=1}^N (y(r) - Y^r)^2 \quad (4.9)$$

donde Y^r es la salida del r -ésimo patrón y $y(r)$ es la salida correspondiente obtenida por la red. Si denotamos por $W = \{w_{ij}\}_{i=1, \dots, n}$ y $\tilde{W} = \{\tilde{w}_{sj}\}_{s=1, \dots, d}$ los vectores de pesos sinápticos asociados a las conexiones hacia adelante y recurrentes respectivamente entonces podemos expresar $E = E(W, \tilde{W})$ luego, en cada iteración del método de la pendiente máxima tendremos

$$W(k + 1) = W(k) + \Delta W(k)$$

$$\tilde{W}(k + 1) = \tilde{W}(k) + \Delta \tilde{W}(k)$$

donde

$$\begin{aligned}\Delta W(k) &= -\alpha \frac{\partial E}{\partial W}(W(k), \tilde{W}(k)) \\ \Delta \tilde{W}(k) &= -\alpha \frac{\partial E}{\partial \tilde{W}}(W(k), \tilde{W}(k))\end{aligned}$$

y α es la constante de aprendizaje de la red. Por otro lado, al derivar la expresión 4.9 se tiene

$$\begin{aligned}\frac{\partial E}{\partial w_{ij}} &= \frac{\partial}{\partial w_{ij}} \left[\frac{1}{2N} \sum_{r=1}^N (y(r) - Y^r)^2 \right] \\ &= \frac{1}{N} \sum_{r=1}^N (y(r) - Y^r) \frac{\partial y(r)}{\partial w_{ij}}.\end{aligned}\quad (4.10)$$

Para determinar las derivadas parciales $\frac{\partial y(r)}{\partial w_{ij}}$ aplicamos la fórmula 1.24 deducida en el capítulo 1:

$$\frac{\partial y(r)}{\partial w_{ij}} = \frac{\partial^e y(r)}{\partial w_{ij}} + \sum_{s=1}^d \frac{\partial^e y(r)}{\partial y(r-s)} \frac{\partial y(r-s)}{\partial w_{ij}}.\quad (4.11)$$

El primer sumando de 4.11 se obtiene de derivar la expresión 4.8. También se debe recordar de la expresión 4.3, que f_i no depende de w_{pq} cuando $p \neq i$; por eso:

$$\frac{\partial^e y(r)}{\partial w_{ij}} = \frac{\partial f_i}{\partial w_{ij}} = \mu_{ij}(x_i(r)).\quad (4.12)$$

Para la sumatoria en 4.11, se debe observar, a partir de 4.8 que

$$\frac{\partial^e y(r)}{\partial y(r-s)} = \tilde{f}'_s(y(r-s)).\quad (4.13)$$

Luego, al reemplazar 4.12 y 4.13 en 4.11 se obtiene:

$$\frac{\partial y(r)}{\partial w_{ij}} = \mu_{ij}(x_i(r)) + \sum_{s=1}^d \tilde{f}'_s(y(r-s)) \frac{\partial y(r-s)}{\partial w_{ij}}\quad (4.14)$$

Por lo tanto:

$$\frac{\partial E}{\partial w_{ij}} = \frac{1}{N} \sum_{r=1}^N (y(r) - Y^r) \left[\mu_{ij}(x_i(r)) + \sum_{s=1}^d \tilde{f}'_s(y(r-s)) \frac{\partial y(r-s)}{\partial w_{ij}} \right].\quad (4.15)$$

Análogamente

$$\frac{\partial E}{\partial \tilde{w}_{sj}} = \frac{1}{N} \sum_{r=1}^N (y(r) - Y^r) \left[\tilde{\mu}_{sj}(y(r-s)) + \sum_{\sigma=1}^d \tilde{f}'_{\sigma}(y(r-\sigma)) \frac{\partial y(r-\sigma)}{\partial \tilde{w}_{sj}} \right].\quad (4.16)$$

Capítulo 5

Resolución Numérica del Problema y Resultados

5.1. Obtención de datos y región de estudio

Los datos para la ejecución del modelo han sido proporcionados por el Servicio Nacional de Meteorología e Hidrología del Perú (SENAMHI) correspondientes a los niveles y caudales promedios diarios desde enero del 2010 a setiembre del 2016, de las Estaciones Hidrométricas San Mateo (a 3210 msnm, latitud $11^{\circ}45'36.9''$ y longitud $76^{\circ}18'4''$), Río Blanco (a 3495 msnm, latitud $11^{\circ}44'17''$ y longitud $76^{\circ}15'20''$) y Chosica (a 906 msnm, latitud $11^{\circ}55'47.5''$ y longitud $76^{\circ}41'22.8''$) mostradas en la figura 5.1.

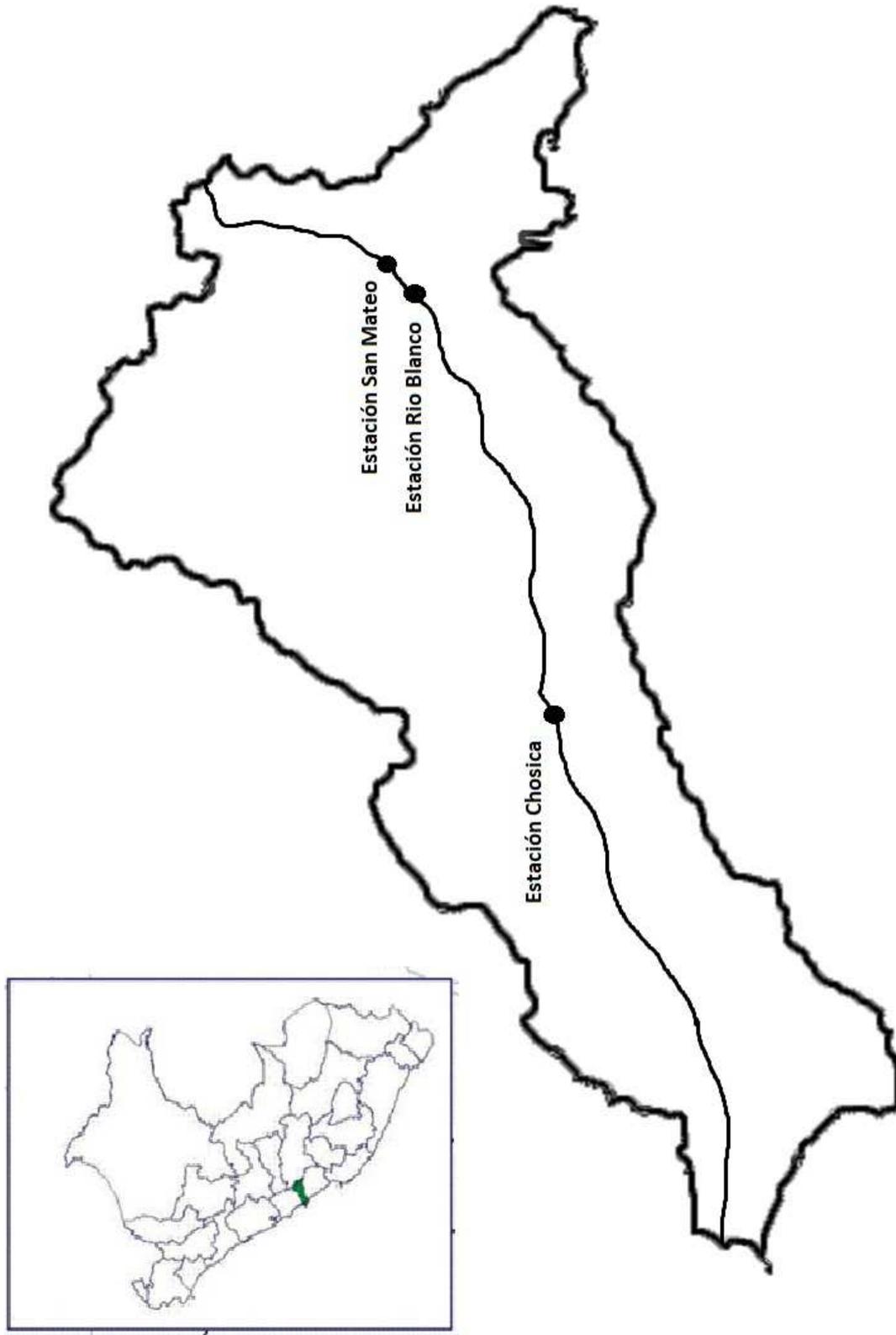


Figura 5.1: Cuenca del río Rimac



Figura 5.2: Estación San Mateo



Figura 5.3: Estación Río Blanco



Figura 5.4: Estación Chosica

5.2. Estructura Neuronal Difusa usada para el modelamiento.

Se ha usado una estructura neuronal difusa como la mostrada en la figura 5.5, donde

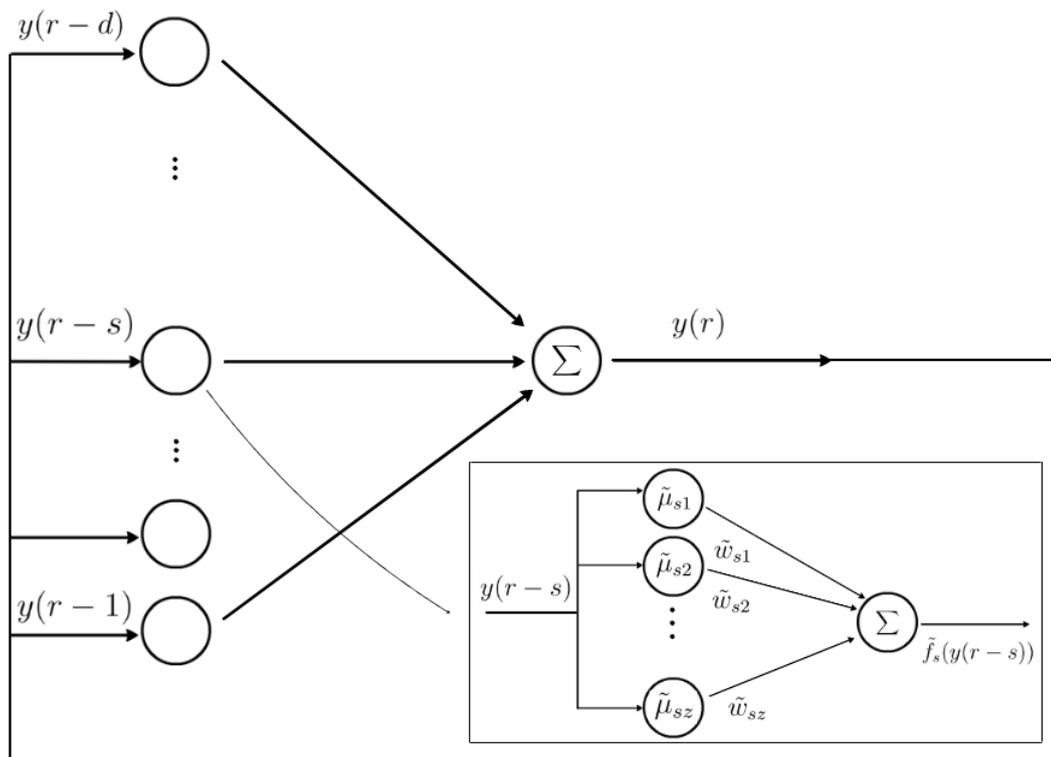


Figura 5.5: Red Neuronal Difusa Recurrente usada para el modelamiento

se aprecia que solo se han usado conexiones recurrentes. Las entradas corresponden a los valores anteriores de los niveles (o caudales) en los días anteriores y la red proporcionará los valores de los niveles (o caudales) en los días posteriores. El modelo original de esta red neuronal difusa ha sido presentado por Altamiranda y Colina en ([16]).

5.3. Implementación y Resultados.

Se ha ejecutado el modelo para la predicción de los caudales del río Rimac medidos en las estaciones San Mateo y Río Blanco, mientras que la predicción de niveles se ha hecho para las estaciones San Mateo, Río Blanco y Chosica.

Es importante notar que el funcionamiento del algoritmo depende del número de funciones de pertenencia (n) usadas, las cuales a su vez dependen del intervalo $[x_{\text{mín}}, x_{\text{máx}}]$ de valores que la variable puede tomar (ver figura 4.2). Mientras más grande sea dicho intervalo, mayor debe ser el número de funciones de pertenencia a considerar. Los cuadros 5.1 y 5.2 muestran el intervalo de valores que (según los registros hidrométricos) han tomado las variables nivel y caudal en las estaciones hidrométricas correspondientes.

	San Mateo	Río Blanco
mín	7.589	0.409
máx	29.684	9.305

Cuadro 5.1: Valores máximo y mínimo del caudal (m^3/s) en las Estaciones San Mateo y Río Blanco.

	San Mateo	Río Blanco	Chosica
mín	2.58	4.30	2.95
máx	12.90	16.00	14.30

Cuadro 5.2: Valores máximo y mínimo del nivel ($dm = 0,1m$) en las Estaciones San Mateo, Río Blanco y Chosica.

Por otro lado, en la etapa de entrenamiento de la red, dependiendo de cada estación hidrométrica, diversos valores han sido asignados a los parámetros: número de patrones(N) y épocas(ep) de entrenamiento, número de iteraciones(k), orden de retraso(d) y horizonte de predicción(H). Dichos valores se muestran en los cuadros 5.3 y 5.4.

	n	N	ep	k	d	H	α
San Mateo	10	5	5	10	4	4	0.1
Río Blanco	12	5	5	10	8	5	0.1

Cuadro 5.3: Valores de los parámetros usados en el modelamiento del caudal.

	n	N	ep	k	d	H	α
San Mateo	10	4	5	20	5	5	0.1
Río Blanco	10	5	4	16	5	4	0.1
Chosica	12	3	5	10	6	5	0.1

Cuadro 5.4: Valores de los parámetros usados en el modelamiento del nivel.

Antes de mostrar los resultados obtenidos, se muestra en la figura 5.6 el diagrama de flujo del entrenamiento de la red:

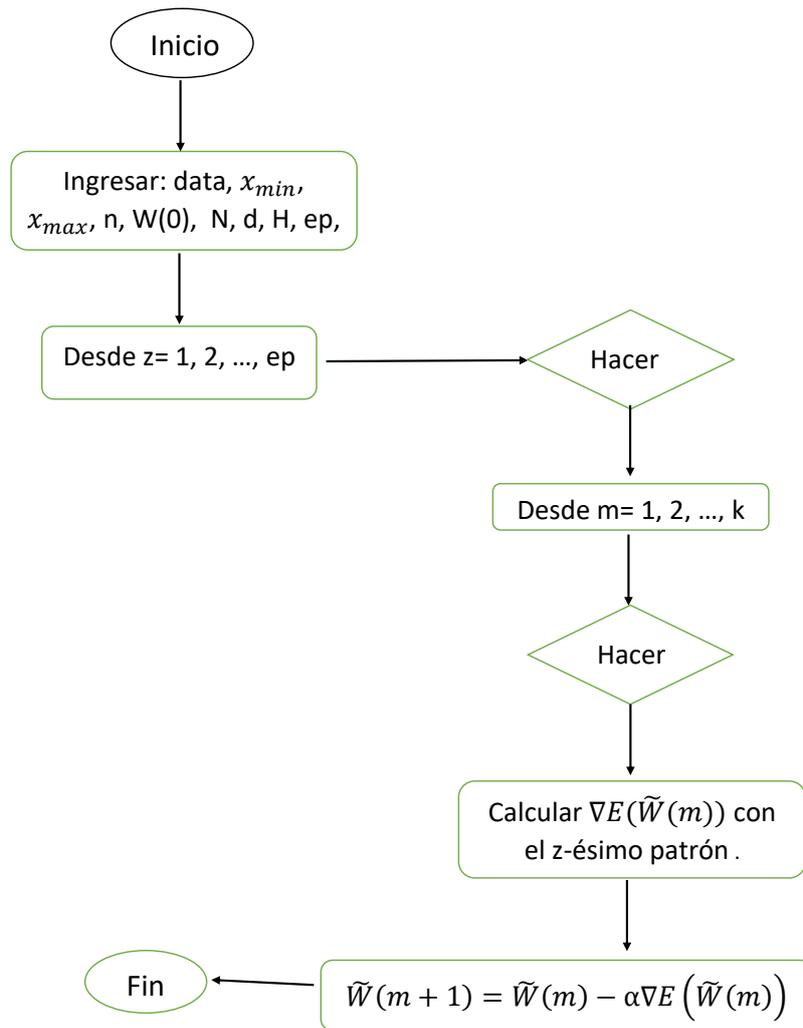


Figura 5.6: Diagrama de flujo

Respecto a la fase de validación, se muestra las gráficas de comparación entre los datos predichos por la red y los datos reales, así como las tablas que muestran los criterios de rendimiento del modelo, tales como el RMSE(root mean square error), MSE(mean square error), MAE(mean absolute error).

	MSE	RMSE	MAE
San Mateo	0.009	0.09	0.07
Río Blanco	0.003	0.05	0.04

Cuadro 5.5: Evaluación del modelamiento del caudal por medio de los errores obtenidos (m^3/s)

Para obtener las siguientes gráficas (Figura 5.7 hasta Figura 5.16), se ha usado como datos de entrada un vector que contiene datos históricos de niveles y caudales (proporcionados por SENAMHI) para cada estación hidrométrica, para primero entrenar la red y luego obtener el vector de errores (Figuras 5.8, 5.10, 5.12, 5.14, 5.16), cuyas componentes deben acercarse a cero, por el método de la pendiente máxima. También se obtiene los pesos óptimos con los cuales haremos funcionar la red y obtener los datos pronosticados para validación con los datos reales (Figuras 5.7, 5.9, 5.11, 5.13, 5.15). Los códigos de los algoritmos usados se pueden ver en el capítulo de anexos.

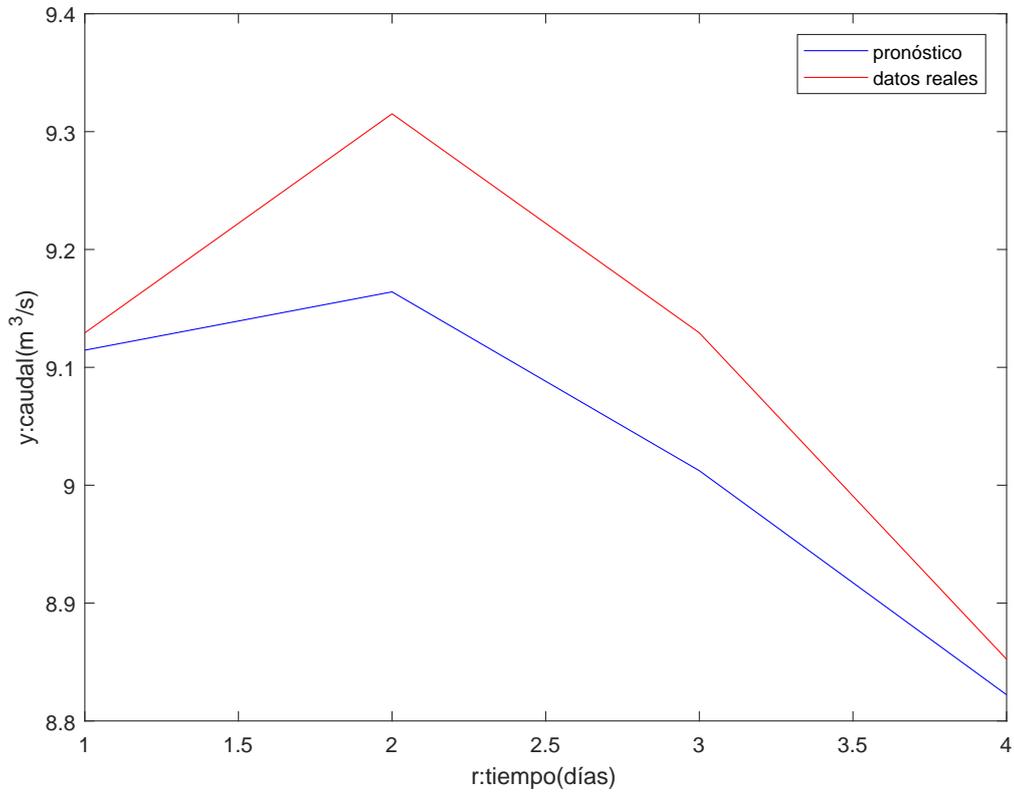


Figura 5.7: Caudal Estación San Mateo

Se observa la cercanía entre los valores de los caudales pronosticados y los valores de los caudales reales, lo cual se ve reflejado en la primera fila de la tabla 5.5.

	MSE	RMSE	MAE
San Mateo	0.32	0.56	0.51
Río Blanco	0.11	0.33	0.25
Chosica	0.07	0.26	0.23

Cuadro 5.6: Evaluación del modelamiento del nivel por medio de los errores obtenidos (*dm*)

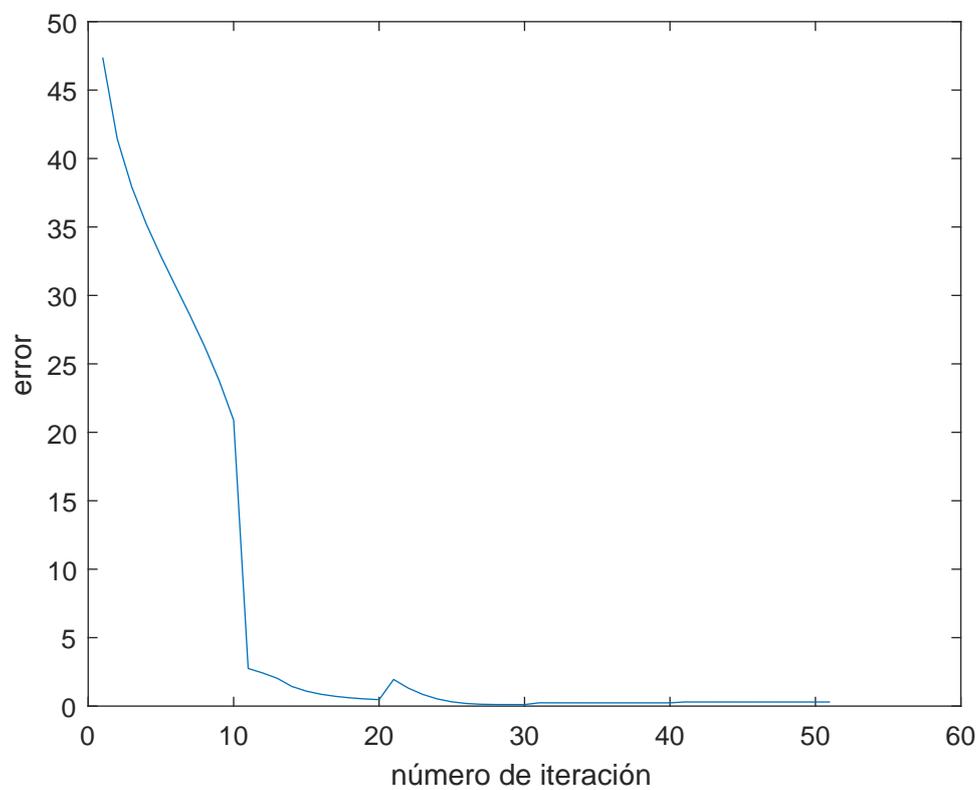


Figura 5.8: Error de entrenamiento. Estación San Mateo

En cada iteración, el error calculado por el método de la pendiente máxima, disminuye de valor.

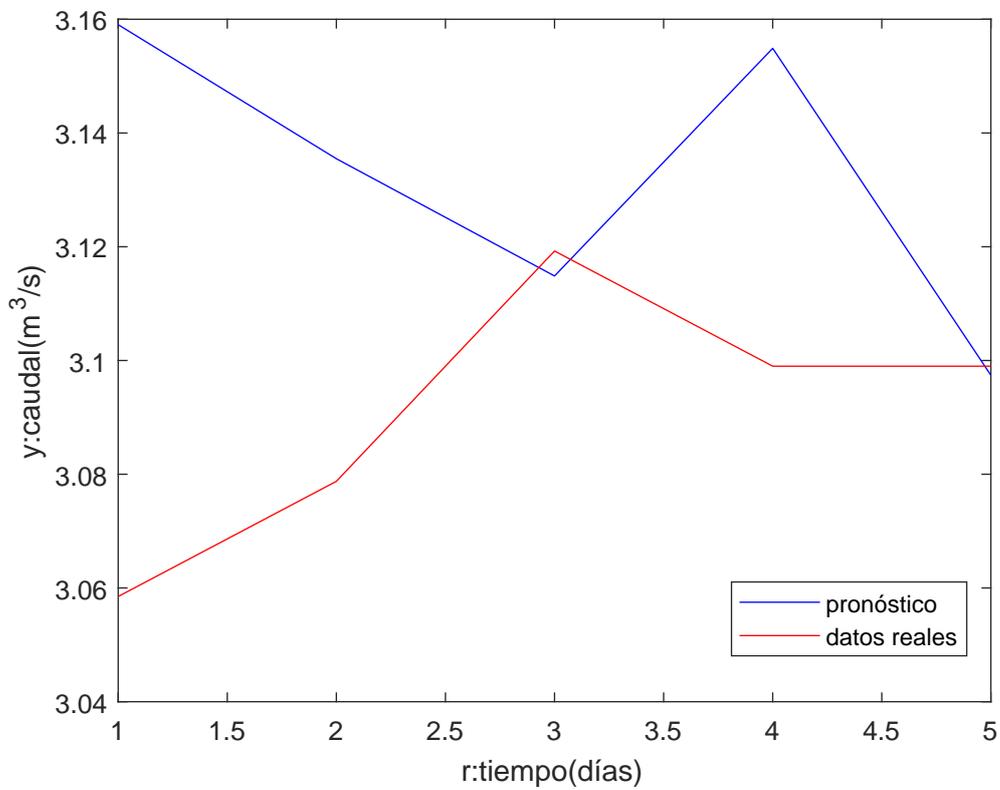


Figura 5.9: Caudal Estación Río Blanco

Además de notar que el error de estimación es pequeño (segunda fila de la tabla 5.5), la función continua que resulta de interpolar los valores estimados, coincide en dos puntos con la respectiva función para los valores reales.

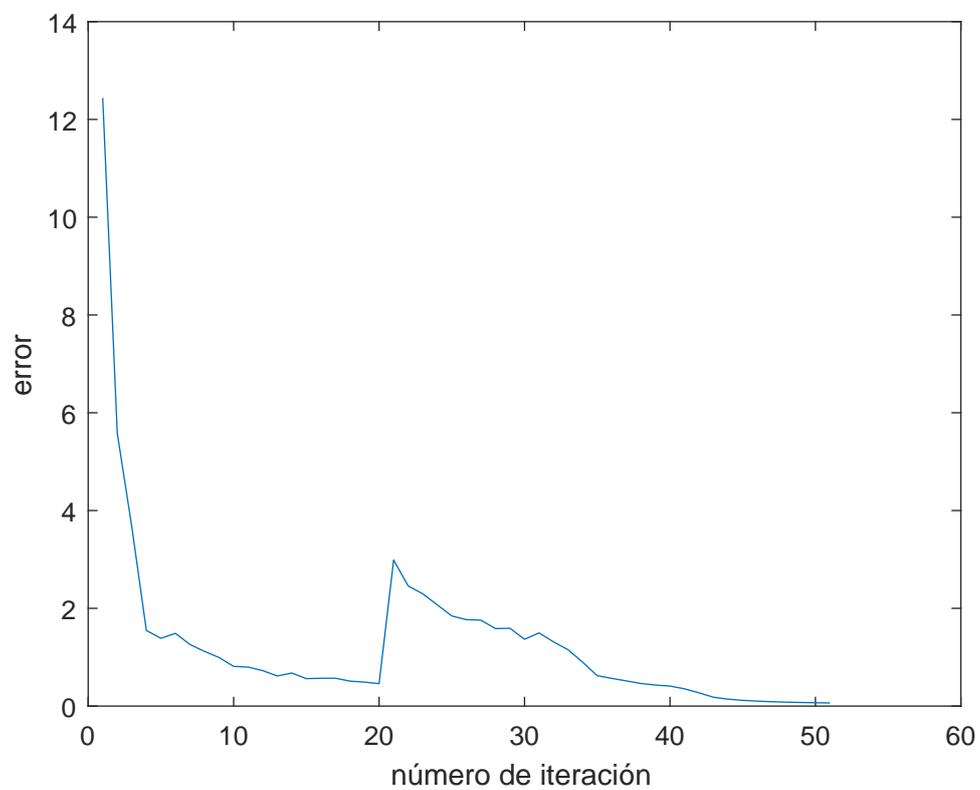


Figura 5.10: Error de entrenamiento. Estación Río Blanco

En cada iteración, el error calculado por el método de la pendiente máxima, disminuye de valor.

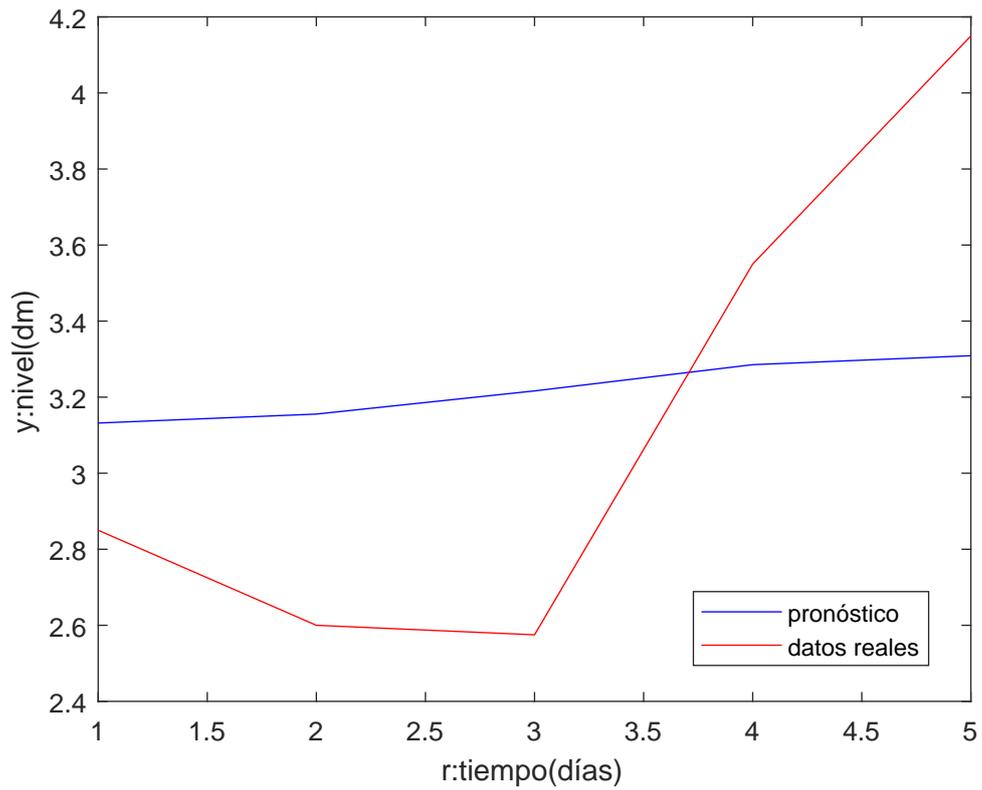


Figura 5.11: Nivel Estación San Mateo

Además de notar que el error de estimación es pequeño (primera fila de la tabla 5.6), la función continua que resulta de interpolar los valores estimados, coincide en un punto con la respectiva función para los valores reales.

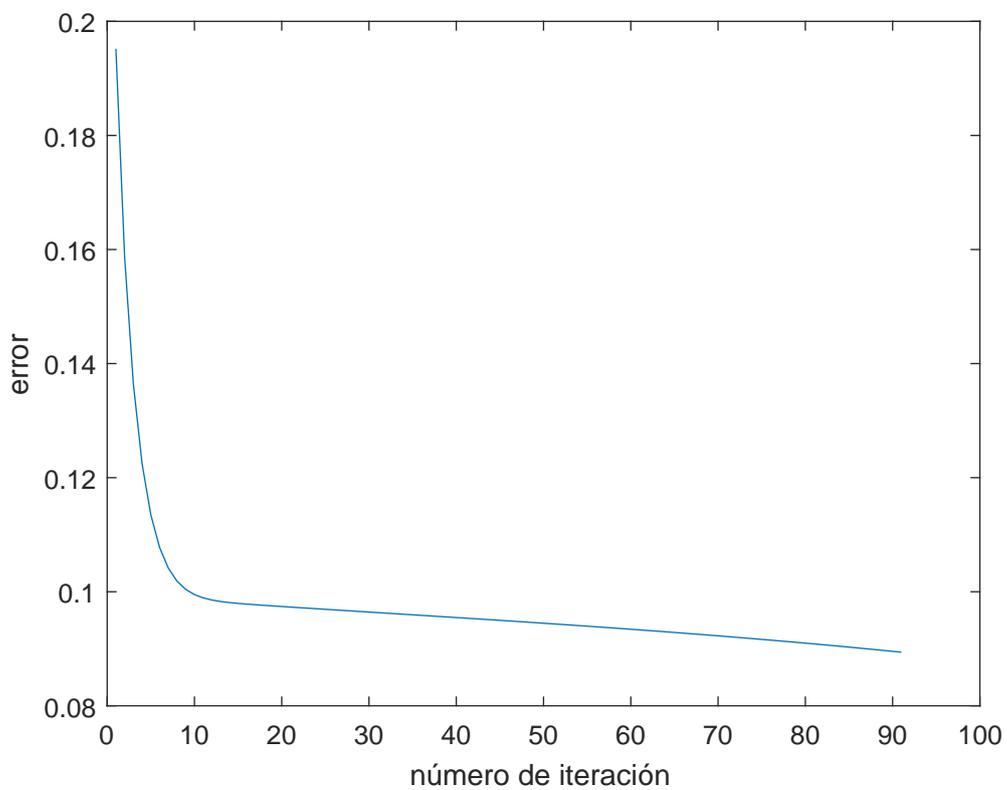


Figura 5.12: Error de entrenamiento. Estación San Mateo

En cada iteración, el error calculado por el método de la pendiente máxima, disminuye de valor.

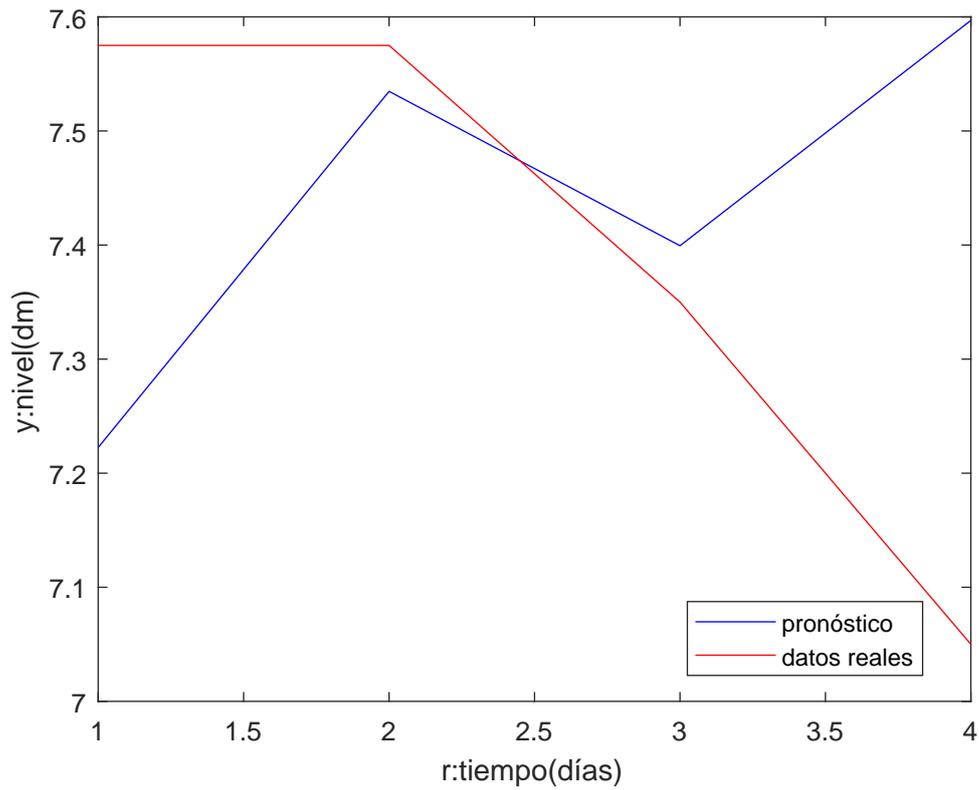


Figura 5.13: Nivel Estación Río Blanco

Además de notar que el error de estimación es pequeño (segunda fila de la tabla 5.6), la función continua que resulta de interpolar los valores estimados, coincide en un punto con la respectiva función para los valores reales.

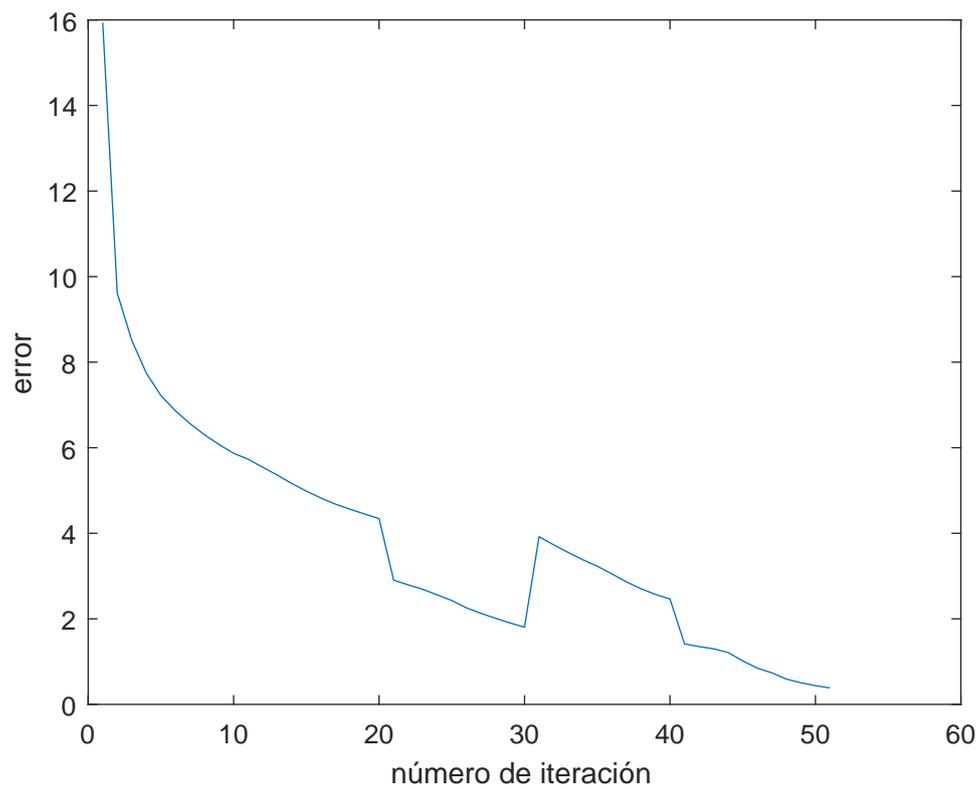


Figura 5.14: Error de entrenamiento. Estación Río Blanco

En cada iteración, el error calculado por el método de la pendiente máxima, disminuye de valor.

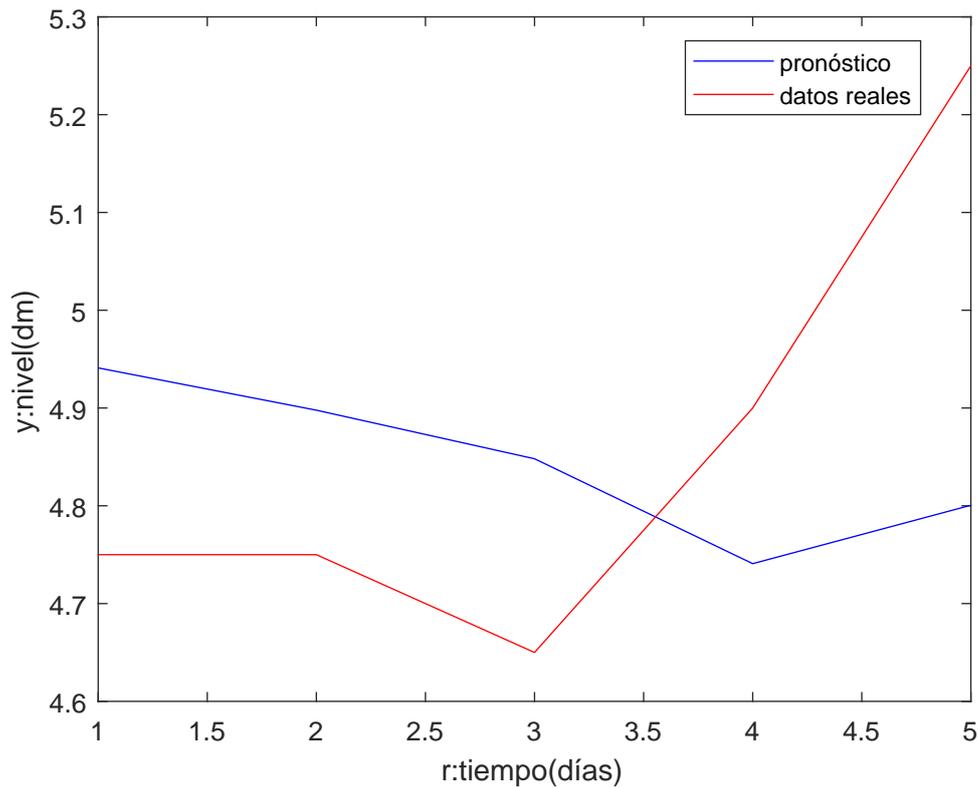


Figura 5.15: Nivel Estación Chosica

Además de notar que el error de estimación es pequeño (tercera fila de la tabla 5.6), la función continua que resulta de interpolar los valores estimados, coincide en un punto con la respectiva función para los valores reales.

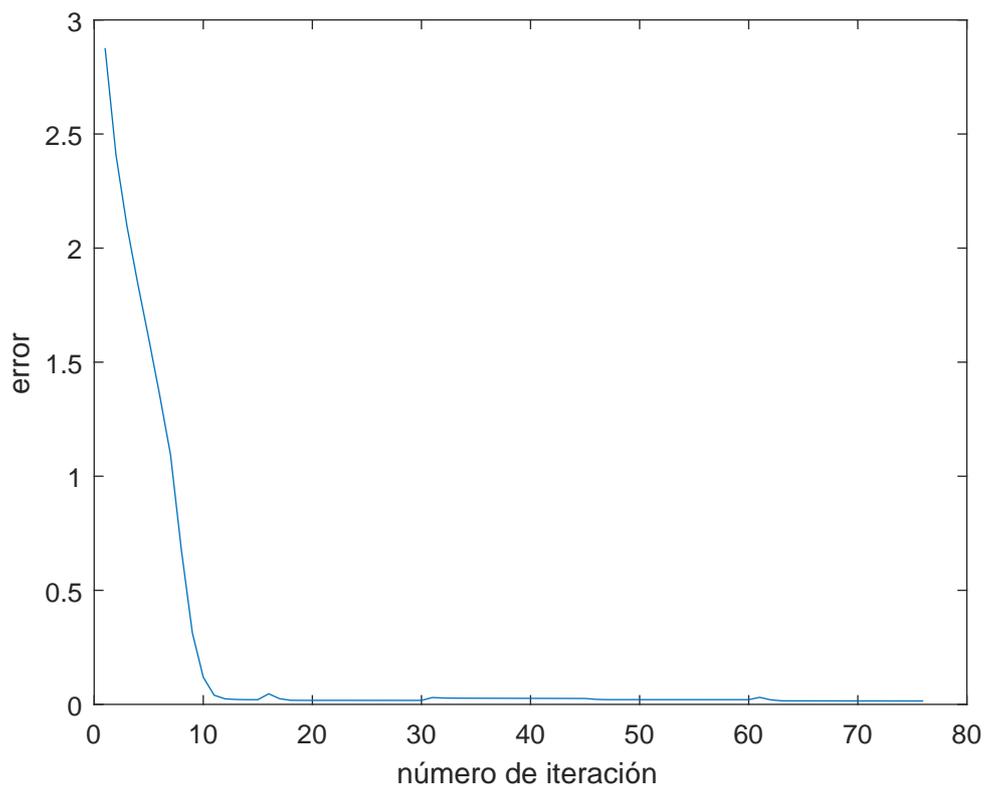


Figura 5.16: Error de entrenamiento. Estación Chosica

En cada iteración, el error calculado por el método de la pendiente máxima, disminuye de valor.

Capítulo 6

Conclusiones

Las figuras 5.8, 5.10, 5.12, 5.14 y 5.16 muestran que el método de la pendiente máxima es útil en este caso para encontrar el mínimo global de la función de error, pues como ya se había mencionado en el capítulo 4, ésta es parabólica respecto a los pesos sinápticos, debido a la elección de funciones de pertenencia triangular.

El número de funciones de pertenencia depende de la longitud del espacio de entrada, el cual ha sido pequeño en todos los casos trabajados en esta tesis (30 como longitud máxima), pero es importante señalar que aunque se disponía del registro hidrométrico del caudal del río Rímac en la estación Chosica, no se ha modelado dicha variable debido a que el espacio de entrada era muy grande en ese caso (longitud 140), ésta situación forzaba a trabajar con más de 50 funciones de pertenencia, lo cual representaba un problema de tiempo computacional. Se espera solucionar dicha dificultad en posteriores trabajos.

En esta tesis se ha mostrado la aplicabilidad de una red neuronal difusa para el modelamiento del caudal y nivel del flujo de agua en un río. Los resultados mostrados en la sección anterior, demuestran que a corto plazo (4 o 5 días), el modelo es suficientemente aproximado como para ser tomado en cuenta con fines de previsión y manejo del recurso hídrico.

La principal ventaja del modelo presentado en este trabajo es que no se necesita un amplio monitoreo de la zona geográfica, ni datos adicionales sobre lluvia u otras variables climáticas; por lo tanto representa una buena alternativa adicional a los modelos

utilizados actualmente para predicción a corto plazo usados por SENAMHI.

Aunque el modelo funciona para predicciones a corto plazo, se espera, en trabajos futuros, extender el horizonte de predicción a mediano y largo plazo. Ello podría conducir a la producción de la tesis de maestría o un artículo.

Anexos

Para cada valor de j ingresado, el siguiente código produce la j -ésima función de pertenencia triangular a usar.

```
function y=MMu(j,x,xmin,xmax,nn)%es la j-esima funcion de pertenencia triangular.
%nn= el numero de funciones de pertenencia triangular a usar. Notar que esto
%determina la regla de correspondencia de Mu.j
%xmin= el valor minimo que x puede admitir
%xmax= el valor maximo que x puede admitir
delta=(xmax-xmin)/(nn-1);
a=0*(abs(x-xmin-delta)*(j-1))>delta)+(1-abs(x-xmin-delta)*(j-1))/delta)*(abs(x-xmin-delta)*(j-1))<=delta);
b=((x-xmax+delta)/delta)*(abs(x-xmax+0.5*delta)<=0.5*delta)+0*(x<(xmax-delta))+(x>xmax);
c=((-x+xmin+delta)/delta)*(abs(x-xmin-0.5*delta)<=0.5*delta)+0*(x>(xmin+delta))+(x<xmin);
y=(abs(j-0.5*(1+nn))<0.5*(nn-1))*a+(j==nn)*b+(j==1)*c;
end
```

Para cada valor de j ingresado, el siguiente código produce la derivada de la j -ésima función de pertenencia triangular a usar.

```
function s=DDmu(j,x,xmin,xmax,nn)%es la derivada de la j-esima funcion de pertenencia triangular.
%nn= el numero de funciones de pertenencia triangular.
%xmin= el valor minimo que x puede admitir
%xmax= el valor maximo que x puede admitir
delta=(xmax-xmin)/(nn-1);
a=(abs(x-xmin-j*delta+1.5*delta)<0.5*delta)<0.5*delta)-(abs(x-xmin-j*delta+0.5*delta)<0.5*delta);
b=-abs(x-xmin-0.5*delta)<0.5*delta);
c=(abs(x-xmax+0.5*delta)<0.5*delta);
s=(abs(j-0.5*(1+nn))<0.5*(nn-1))*a+(j==1)*b+(j==nn)*c;
```

end

Para cada valor de x ingresado, el siguiente código produce un vector cuyas componentes son las derivadas de cada una de las funciones de pertenencia triangular evaluadas en el punto x .

```
function A=DDdmu(x,xmin,xmax,nn) %es el vector de derivadas de las funciones de pertenencia triangular.
%la j-esima componente de este vector es la derivada de la j-esima funcion
%triangular evaluada en x
%nn= el numero de funciones de pertenencia triangular. Notar que esto
%determina la regla de correspondencia de MMuj
%xmin= el valor minimo que x puede admitir
%xmax= el valor maximo que x puede admitir
A=zeros(1,nn);
for j =1:nn
    A(j)=DDdmu(j,x,xmin,xmax,nn);
end
end
```

El siguiente código reproduce el funcionamiento de nuestra red neuronal difusa, es decir, recibe como entrada a un vector x de datos iniciales (niveles o caudales) cuya longitud es igual a dd (orden de retraso) y produce como salida un vector de longitud H (horizonte de predicción) que contiene los datos predichos por la red.

```

function y=RRnet(xmin, xmax, nn, x, w, dd, H)
%genera un vector de salidas de longitud H cuando
%nn= el numero de funciones de pertenencia triangular.
%w=pesos (matriz de dimension dd x nn);
%xmin= el valor minimo que x puede admitir
%xmax= el valor maximo que x puede admitir
%a=data inicial (un vector de longitud igual a dd)
%H=Horizonte de prediccion
%dd=orden de retraso
y=zeros(1,H);
xe=[x y];
for r=1:H
    P=zeros(dd,nn);
    for j=1:nn
        for i=1:dd
            P(i,j)=w(i,j)*MMu(j, xe(dd+r-i), xmin, xmax, nn);
        end
    end
    y(r)=(sum(sum(P)));

```

```
xe(dd+r)=y(r);  
end  
end
```

El siguiente código recibe un patrón de datos entrada-salida y produce el error de predicción de nuestra red neuronal difusa con respecto a dicho patrón.

```
function ee=errorpattern(xmin,xmax,nn,dd,pattern,w,H,N)  
ee=0;  
for r=1:N  
x=pattern(r,1:dd);  
yd=pattern(r,dd+1:dd+H);  
y=RRnet(xmin,xmax,nn,x,w,dd,H);  
ee=ee+norm(y-yd)^2;  
end  
ee=ee/(2*N*H);
```

$$\frac{\partial y(r)}{\partial w_{ij}}$$

El siguiente código recibe un vector de datos entrada-salida y produce una matriz tridimensional cuyos elementos son las derivadas parciales de la expresión 4.11.

```

function dY=GGGradd(xmin,xmax,nn,x1,x2,w,dd,H)
%calcula la derivadas parciales dy(beta)/w_{ij}= dY(i,j,beta)
%con beta=1,2,...,H
%k1=data inicial, osea un vector de longitud dd.
%k2=las siguientes componentes,(vector de longitud H)
%generalmeente x2=RRnet(xmin,xmax,m,x1,w,dd,H)
%donde H seria el horizonte prediccion para la cual sera entrenada
%a red.
%w= pesos (matriz de tamaño dd x nn)
%dd=orden de retraso
%nn= numero de funciones de pertenencia(es la misma en cada neurona de entrada)
dY=zeros(dd,nn,H);
deY=zeros(dd,nn,H);
x=[x1 x2];
for beta=1:H
    for j=1:nn
        for i=1:dd
            deY(i,j,beta)=MMu(j,x(dd+beta-i),xmin,xmax,nn);
        end
    end
end

```

```

end
for i=1:dd
    for j=1:nn
        A=zeros(H,dd+1);
        A(:,1)=deY(i,j,:);
        dY(i,j,1)=deY(i,j,1);
        for alpha=2:H
            for tau=2:min([alpha dd+1])
                A(alpha,tau)= dY(i,j,alpha-tau+1)*(dot(w(tau-1,:),DDmuu(x(dd+alpha-tau+1),xmin,xmax,nn)));
            end
        end
        dY(i,j,alpha)=sum(A(alpha,:));
    end
end
end
end

```

El siguiente código calcula las derivadas parciales $\frac{\partial E}{\partial w_{ij}}$ cuya fórmula está dada en la expresión 4.10.

```

function dE=EEgradd(xmin,xmax,nn,pattern,w,dd,H,N)
%dd=orden de retraso
%nn= numero de funciones de pertenencia de cada capa
%N=numero de patrones de entrenamiento de cada epoca.
%H=horizonte de prediccion de la red
%de es una matriz tamaño dd x nn cuya entrada ij es la derivada dE/dw_{ij}
%pattern es una matriz de tamaño Nx (dd+H) cuyas filas son los patrones de

```

```

%entrada(vector de longitud dd) – salida(vector de longitud H)para entrenar la red.
%general, x2=RRnet(xmin, xmax, nn, x1, w, dd, N)
%w= pesos (matriz de tamaño dd x nn)
dE=zeros(dd, nn);
for i =1:dd
    for j =1:nn
        EE=zeros(1, N);
        for r =1:N
            x1=pattern(r, 1:dd); %entrada del r-esimo patron
            x2=RRnet(xmin, xmax, nn, x1, w, dd, H); %salida de la red(r-esimo patron)
            yd=pattern(r, dd+1:dd+H); %salida deseada (r-esimo patron)
            errorvec=x2-yd; % vector de error(salida red menos salida deseada)
            dY=GGGradd(xmin, xmax, nn, x1, x2, w, dd, H);
            ddY=reshape(dY(i, j, :), [1 H]);
            EE(r)=dot(errorvec, ddY);
        end
    end
    dE(i, j)=sum(EE)/(N*H);
end
end
end

```

El siguiente código transforma un vector de patrones de entrada a usar en una época de entrenamiento en una matriz para que sea más fácil de usar en el código de entrenamiento de la red.

```
function pattern=transform(dataep, delta, dd, N, H)
```

```

%transforma el vector dataep en una matriz de orden N x (dd+H)
%para poder ser usada en una epoca del training de la red.
L=length ( dataep );
pattern=zeros (N,dd+H);
for r=1:N
    pattern (r,:)=dataep (L-dd-H-(N-r)*delta+1:L-(N-r)*delta );
end

```

El siguiente código realiza el entrenamiento de la red cuando se ingresa los patrones de entrenamiento, el número de épocas, números de iteraciones en cada época y una matriz de pesos iniciales de donde se empezará a usar el método de la pendiente máxima. Como salida, produce el vector de errores cuyas gráficas se han mostrado en el capítulo 5 (Figuras 5.8, 5.10, 5.12, 5.14, 5.16). También produce la matriz W_{opt} , de pesos óptimos a usar en la red para la etapa de validación, y son los pesos con los que se han obtenido las salidas pronosticadas y con dichas salidas se ha elaborado las gráficas dadas en el capítulo 5 (Figuras 5.7, 5.9, 5.11, 5.13, 5.15)

```

function [ errortraining , Wopt]=train (xmin ,xmax ,nn ,datafortraining , delta ,w0 ,H,N ,iternum , eptime , alpha )
%errortraining=vector de longitud iternum*epnum+1 que contiene
%a todos los errores de cada iteracion.
%Wopt=pesos optimos
%epnum= numero de epocas de entrenamiento
%iternum=numero de iteraciones en cada epoca
%w0=pesos iniciales
%alpha=constante de aprendizaje de la red

```

```

W=zeros(dd,nn,iternum*epnum+1);
W(:, :, 1)=w0;
%L=length( datatotal );
%datafortesting=datatotal(L-H+1:L);
Delta=dd+H+(N-1)*delta;
E=zeros(1,iternum*epnum);
%datafortraining=datatotal(1:L-H);
LT=length( datafortraining );
for z=1:epnum
    dataep=datafortraining(LT-(epnum-z+1)*Delta+1:LT-(epnum-z)*Delta);
    pattern=transform( dataep, delta, dd, N, H );
    for k=(z-1)*iternum+1:z*iternum
        E(k)=errorpattern( xmin, xmax, nn, dd, pattern, W(:, :, k), H, N );
        dE=EEgradd( xmin, xmax, nn, pattern, W(:, :, k), dd, H, N );
        W(:, :, k+1)=W(:, :, k)-alpha*dE;
    end
end
Wopt=W(:, :, epnum*iternum+1);
E(epnum*iternum+1)=errorpattern( xmin, xmax, nn, dd, pattern, W(:, :, epnum*iternum+1), H, N );
errortraining=E;
%plot(1:epnum*iternum+1,E, ' * ' )

```

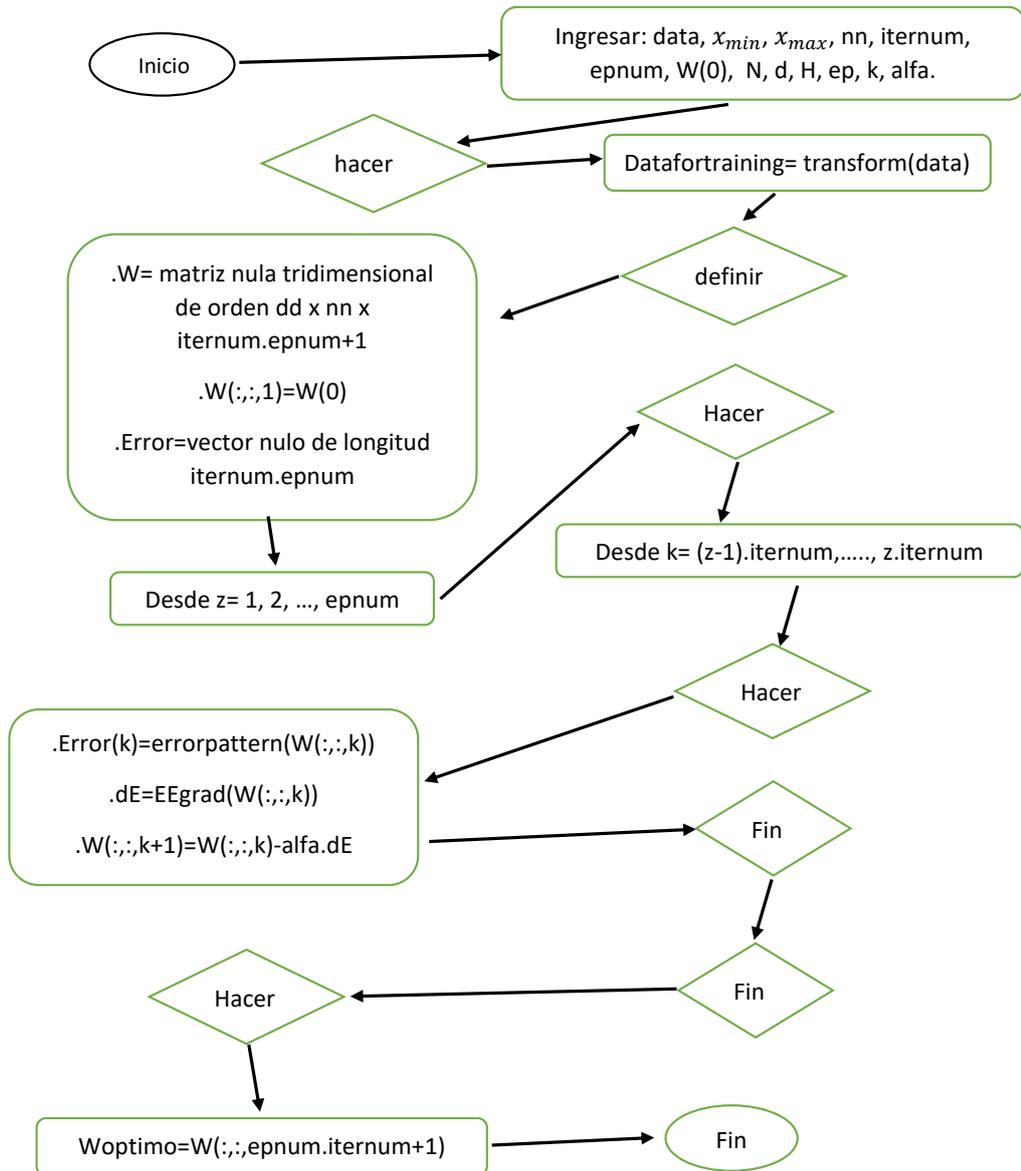


Figura 6.1: Diagrama de flujo para obtener los pesos óptimos al entrenar la red

Bibliografía

- [1] NOCEDAL J., STEPHEN J., *Numerical Optimization* , Springer, 1999.
- [2] ELLACOTT S., MASON J. y ANDERSON I., *Mathematics of Neural Networks. Models, Algorithms and Applications*, Springer, 1997.
- [3] GUPTA, M.M. AND SINHA, N.K. AND IEEE NEURAL NETWORKS COUNCIL.,
- [4] ROJAS R., *Neural Networks. A Systematic Introduction*, Springer, 1996.
- [5] LESHNO, M. AND SCHOCKEN, S., Multilayer Feedforward Networks with Non-Polynomial Activation Functions Can Approximate Any Function, *Information Control*, **8**, págs. 338–353, 1965.
- [6] ZADEH, L., Fuzzy Sets, *Working paper (ha-Makhon le-Mehkar 'Asakim be-Yisra'el)*, **6** págs. 861–867, 1991.
- [7] ZADEH, L., The concept of a linguistic variable and its application to approximate reasoning-I, *Information Sciences*, **8**, págs. 199–249, 1975.
- [8] DUBOIS, D., *Fuzzy sets and systems: theory and applications*, Academic Press, 1980.
- [9] MAGUIÑA, R., Sistemas de inferencia basados en Lógica Borrosa: Fundamentos y caso de estudio , *Revista de Investigación de Sistemas e Informática*, **7**, págs. 91–104, 2010.
- [10] MAMDANI, E., Application of fuzzy logic to approximate reasoning using linguistic synthesis, *IEEE Computer Society Press*, págs. 196–202, 1976.
- [11] TAKAGI, T. AND SUGENO, M., Fuzzy Identification on Systems and its Applications to Modeling and Control, *IEEE Tran. on Systems*, **15** págs. 116–132, 1985.

- [12] SUGENO, M. AND KANG, G., Structure Identification of Fuzzy Model, *Fuzzy Sets and Systems*, **28** págs. 329–346, 1986.
- [13] MONIKA, A., Comparison of Mamdani Fuzzy model and Neuro Fuzzy Model for load sensor , *International Journal of Engineering and Innovative Technology (IJEIT)* , **2**, Issue 9. 2013.
- [14] YAMAKAWA, T., A Neo Fuzzy Neuron and its Application to System Identification and Prediction of Chaotic Behavior *IEEE Press*, págs. 383–405, 1994.
- [15] FERNÁNDEZ N., JAIMES W., ALTAMIRANDA E., Neuro-fuzzy modeling for level prediction for the navigation sector on the Magdalena River (Colombia) , *Journal of Hydrodynamics*, **12**, págs. 36–50, 2010.
- [16] ALTAMIRANDA, E. y COLINA E, A Recurrent Fuzzy Neuron for Online Modeling of Nonlinear Systems , *Proceedings of 15th Triennial IFAC World Conference*, **8**, Barcelona, 2002.
- [17] PINKUS A., Approximation Theory of the MLP model in Neural Networks , *Acta Numerica*, págs. 143–195, 1999.
- [18] BIELECKI A, Mathematical Model of Architecture and Learning Process of Artificial Neural Networks , *Tasks Quarterly*, **7**, págs. 193–114, 2003.
- [19] MOYO V., SIBANDA K., On the Optimal Learning Rate Size for the Generalization Ability of Artificial Neural Networks in Forecasting TCP/ICP Traffic Trends , *International Journal of Computer Applications* , **105**, págs. 09–14, 2014.
- [20] NORTHOVER F., On the Method of Steepest Descents , *Journal of Mathematical Analysis and Applications*, **29**, págs. 216–232, 1970.
- [21] FRENCH M. N., KRAJEWSKI W. F. y Y CUYKENDALL R., Rainfall forecasting in space and time using a neural network , *Journal of Hydrology, Amsterdam*, **137**, págs. 1–31, 1992.
- [22] CHANG I., RAPIRAJU S., WHITESIDE M., y HWANG G, A neural network to time series forecasting , *Proceedings of the Decision Science Institute*, **3**, págs. 1716–1718, 1991.

- [23] DANIELL T. M., Neural networks, applications in hydrology and water resources engineering , *Proceedings of the International Hydrology and Water Resources Symposium*, **3**, págs. 797–802, 1991.
- [24] KARUNANITHI N., GRENNY WILLIAM J., WHITLEY D y BOVEE KEN, Neural networks for river flow prediction , *Journal of Computing in Civil Engineering* , **8**, págs. 201–220, 1994.
- [25] ANA. DGCRH.(2012). Resultado del monitoreo de la calidad del agua de la cuenca del río Rímac. Informe técnico N° 006-2012.