Análisis de la Robustez de un Modelo de Redes Neuronales para la Predicción de Caudales en la cuenca alta del río Paute

Análisis de la Robustez de un Modelo de Redes Neuronales para la Predicción de Caudales en la cuenca alta del río Paute

LENIN VLADIMIR CAMPOZANO PARRA
Ingeniero Civil
Egresado de Maestría en Métodos Numéricos para Diseño en Ingeniería
Unidad de Posgrados
Universidad Politécnica Salesiana

Dirigido por:
ESTEBAN SAMANIEGO ALVARADO
Ingeniero Civil
Magister en Métodos Numéricos para Cálculo y Diseño en Ingeniería
PhD. en Mecánica Computacional
Unidad de Posgrados.
Universidad Politécnica Salesiana

Cuenca –Ecuador.

Da tos de Catalogación Bibliográfica.

CAMPOZANO PARRA LENIN VLADIMIR

Análisis de la Robustez de un Modelo de Redes Neuronales para la Predicción de Caudales en la cuenca alta del río Paute

Universidad Politécnica Salesiana, Cuenca – Ecuador, 2011
MAESTRÍA EN MÉTODOS NUMÉRICOS PARA DISEÑO EN INGENIERÍA
Formato 170 x 240 Páginas: 121

Breve reseña del autor e información de contacto:

Lenin Vladimir Campozano Parra
Ingeniero Civil.
Egresado de la Maestría en Métodos Numéricos para Diseño en Ingeniería.
Unidad de Posgrados.
Universidad Politécnica Salesiana del Ecuador.
lenin_camp@yahoo.com

Dirigido por:

Esteban Samaniego Alvarado.
Ingeniero Civil
Magister en Métodos Numéricos para Cálculo y Diseño en Ingeniería
PhD. En Mecánica Computacional
Catedrático de la Maestría en Métodos Numéricos de la Universidad Politécnica Salesiana del Ecuador
esteban.samaniego@gmail.com

Todos los derechos reservados.
Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución, comunicación pública y transformación de esta obra para fines comerciales, sin contar con autorización de los titulares de propiedad intelectual. La inflación de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual. Se permite la libre difusión de este texto con fines académicos o investigativos por cualquier medio, con la debida notificación a los autores.

DERECHOS RESERVADOS
©2011 Universidad Politécnica Salesiana.
CUENCA - ECUADOR – SUDAMÉRICA.

CAMPOZANO PARRA LENIN VLADIMIR
Análisis de la Robustez de un Modelo de Redes Neuronales para la Predicción de Caudales en la cuenca alta del río Paute.
Edición y Producción:
Patricia Guamán integratehoy@hotmail.com
Diseño de la portada:
Lenin Vladimir Campozano Parra

IMPRESO EN ECUADOR - PRINTED IN ECUADOR.

ÍNDICE GENERAL

Contenido

ÍNDICE GENERAL 5
ÍNDICE DE FIGURAS 9
ÍNDICE DE TABLAS 11
DEDICATORIA 13
PREFACIO. 14
PROLOGO. 15
AGRADECIMIENTOS. 16
PARTE 1 17
CAPÍTULO 1 19
INTRODUCCIÓN 19
1.1 Generalidades 19
1.2 Motivación 21
1.3 Objetivos 21
1.3.1 Objetivo general 21
1.3.2 Objetivos específicos 21
1.4 Esquema de tesis 22
CAPÍTULO 2 27
REDES NEURONALES ARTIFICIALES 27
2.1 Introducción 27
2.2 Neuronas con argumento escalar 28
2.3 Funciones de transferencia 29
2.3.1 Función Escalón o limitador fuerte 29
2.3.2 Función lineal 29

2.3.3 Función sigmoidal 30
2.3.4 Resumen de funciones de transferencia 30
2.4 Neuronas con argumento vectorial 31
2.5 Topología de una red neuronal 32
2.6 Arquitecturas de red 33
2.6.1 Multilayer Perceptron 34
2.6.2 Radial Basis Networks (RBN) 34
2.6.3 Redes de Aprendizaje Competitivo (CLN) 37
2.6.4 Committee Machines 40
2.7 Aprendizaje 40
2.7.1 Algoritmo Levenberg-Marquardt 41
2.7.2 Algoritmo Backpropagation 43
2.7.3 Métodos basados en gradientes conjugados 45
CAPÍTULO 3 49
DEFINICIÓN DE LA ROBUSTEZ DE UN MODELO 49
3.1 Definición de Robustez del modelo 49
3.2 Análisis de sensibilidad 49
3.2.1 Cálculo de la sensibilidad 50
3.3 Análisis de incertidumbre 51
3.3.1 Un método probabilístico para generación de la incertidumbre 52
3.3.2 Propagación de la incertidumbre por la descomposición de la varianza 54
PARTE 2 57
CAPITULO 4 59
PRE PROCESAMIENTO DE LOS DATOS 59
4.1 Definición de los datos 59
4.2 Escala de datos 62

4.3 Identificación de variables de entrada 63
4.4 Análisis de resultados 67
CAPÍTULO 5 71
IMPLEMENTACIÓN DE ARQUITECTURAS DE REDES NEURONALES 71
5.1 Toolbox de Matlab 71
5.2 Tipos de redes estudiadas 72
5.3 Identificación de la red optima 77
5.3.1 Red una capa oculta sigmoidal y capa de salida sigmoidal 78
5.3.2 Red una capa oculta sigmoidal y capa de salida lineal 79
5.3.3 Red dos capas ocultas sigmoidales y capa de salida lineal 81
5.3.4 Red tres capas ocultas sigmoidales y capa de salida lineal 81
5.3.5 Determinación de la red optima 82
5.4 Comparación de predicción con Software Comercial 84
5.5 Análisis de eventos extremos 85
5.6 Análisis de resultados 89
CAPÍTULO 6 93
ROBUSTEZ DEL MODELO PROPUESTO 93
6.1 Sensibilidad del modelo a las estaciones 93
6.2 Análisis de Incertidumbre 94
6.2.1 Incertidumbre del modelo en predicción 95
6.2.2 Propagación de incertidumbre en medición 96
6.3 Análisis de Resultados 101
CAPÍTULO 7 105
CONCLUSIONES Y TRABAJO A FUTURO 105
7.1 Recapitulación 105
7.2 Conclusiones 107
7.3 Trabajo a futuro 108

BIBLIOGRAFÍA 113

ÍNDICE DE FIGURAS

Figura 1 *Modelo simplificado de una neurona* 27
Figura 2 *Modelo de una Neurona.* 28
Figura 3 *Función de transferencia escalón* 29
Figura 4 *Función de transferencia lineal* 30
Figura 5 *Función de transferencia sigmoidal* 30
Figura 6 *Neurona con argumento vectorial* 32
Figura 7 *Topologías de red.* 32
Figura 8 *Redes multicapas.* 33
Figura 9 *Mapeo de un dominio bidimensional mediante un MLP* 34
Figura 10 *Funciones de activación tipo Radial Basis. Tres ejemplos.* 35
Figura 11 *Aproximación de una función mediante la combinación lineal de funciones radiales* 35
Figura 12 *Red de mapeo lineal* $\boldsymbol{RM} \rightarrow \boldsymbol{RN}$ 38
Figura 13 *Representación de datos agrupados mediante una red Self-Organizing-Map.* 39
Figura 14 *Histograma de los errores de predicción del modelo* 53
Figura 15 *Distribución espacial de las estaciones de precipitación y caudal* 60
Figura 16 *Grafico correlación cruzada de estaciones de precipitación con la de caudal* 64
Figura 17 *Grafico de la auto-correlación del caudal* 65
Figura 18 *Grafico de valores propios* 66
Figura 19 *Predicción red lineal* 72
Figura 20 *Pantalla Interfaz durante entrenamiento de la red 2 capas* 73
Figura 21 *Predicción MLP capa sigmoidal y capa lineal* 74
Figura 22 *Pantalla Interfaz durante entrenamiento de la red MLP 3 capas* 75
Figura 23 *Predicción MLP 2 capas sigmoidal y capa lineal* 75
Figura 24 *Predicción MLP 3 capas sigmoidal y capa lineal* 76
Figura 25 *Predicción MLP 3 capas sigmoidal y capa lineal* 76
Figura 26 *Modelo en entrenamiento, validación y predicción* 77
Figura 27 *Respuesta Q^2 vs número de neuronas red 5.3.1* 79
Figura 28 *Respuesta Q^2 vs número de neuronas red 5.3.2* 80
Figura 29 *Gráfico de predicción de caudales con un día de avance.* 85

Figura 30 *Relación entre caudales observados y calculados menores al percentil 20* 87
Figura 31 *Relación entre caudales observados y calculados mayores al percentil 80* 88
Figura 32 *Relación entre caudales observados y calculados ente el percentil 20 y 80* 89
Figura 33 *Distribución del error en predicción del modelo* 95
Figura 34 *Esquema de propagación de los errores de medición de precipitación* 96
Figura 35 *Propagación de la incertidumbre de la medición de precipitación* 97
Figura 36 *Relación lineal entre la varianza del error de predicción de caudales y la Varianza del error de Precipitación* 99

ÍNDICE DE TABLAS

Tabla 1 *Resumen de funciones de transferencia* 31
Tabla 2 *Estaciones de precipitación y caudal de la cuenca alta del Paute*..... 50
Tabla 3 *Fechas de datos coincidentes de información*..... 59
Tabla 4 *Estaciones de precipitación y caudal, unidades, pesos, ubicación* 61
Tabla 5 *Valores propios de la matriz de covarianza.*..... 66
Tabla 6 *Valores de Q^2 red 5.3.1* 78
Tabla 7 *Valores de Q^2 red 5.3.2* 80
Tabla 8 *Valores de Q^2 red 5.3.3* 81
Tabla 9 *Valores de Q^2 red 5.3.4* 82
Tabla 10 *Valores de Q^2 para varias arquitecturas de redes neuronales* 83
Tabla 11 *Valores del coeficiente de eficiencia para diversos caudales* 86
Tabla 12 *Sensibilidad promedio del modelo a las estaciones* 93
Tabla 13 *Sensibilidad por estación* 94
Tabla 14 *Varianza del error de predicción de caudal para diversas varianzas del error de precipitación*..... 98
Tabla 15 *Incertidumbre total del modelo en predicción* 100

DEDICATORIA

Al Gran Arquitecto Del Universo y Al Dios de mi Corazón

A Mi Padre, mi Hermana Karla y a Mi Hermosa Madre

A mis hijos Dominick, Micaela, Rafaela y David

A mí Amada Esposa Jenny

PREFACIO.

En esta tesis se analiza la robustez del método de las redes neuronales para la predicción de caudales aplicado a la cuenca alta del Rio Paute. El método de las redes neuronales es un modelo basado en los datos o data-driven model. El análisis de la robustez del modelo nos sirve para evaluar las variaciones en la respuesta del modelo debido a variaciones en los datos de entrada y a la capacidad de predicción del modelo. Por esta razón se realiza un análisis de sensibilidad y un análisis de incertidumbre del modelo para estudiar la robustez del mismo.

Del análisis de sensibilidad se busca obtener información sobre la importancia de cada variable en el modelo. Con esta información podemos tomar decisiones orientadas más específicamente. Del análisis de incertidumbre se busca obtener información sobre el nivel de incertidumbre del modelo, específicamente para el problema de predicción de caudales. Como fuentes de incertidumbre se han estudiado la incertidumbre estructural del modelo y la propagación de incertidumbre debido a la incertidumbre en los datos de entrada.

El método de las redes neuronales será utilizado para la predicción de caudales, dados los datos de precipitación e historia del caudal como variables de entrada. Es importante recalcar que los datos se han obtenido por mediación del Grupo de la Tierra y del Ambiente, ante diversas entidades que han monitoreado la cuenca del Rio Paute. Las estaciones de registro de precipitación son: Cancan Soldados, Chanlud, El Portete, El Labrado, Machangara-DJ Chulco, Matadero en Sayausí, Tarqui DJ Cumbe, Ucubamba en ETAPA, Yanuncay en Pucan y la estación de registro de caudal es en el Tomebamba en Ucubamba. Debido a que el modelo de redes neuronales recibe la información de cada una de las estaciones, sin que los datos de precipitación hayan sido concentrados, entonces el modelo estudiado es distribuido y es un modelo basado en los datos. Para la elaboración de un modelo matemático distribuido es necesario información detallada de la cuenca, como tipo de suelo, pendientes del terreno, vegetación, uso del suelo, evaporación, evapotranspiración, etc., lo cual hace complicada su implementación debido a la falta de estos parámetros. El uso de redes neuronales para la elaboración de un modelo distribuido tiene la ventaja de la independencia de estos factores debido a que la red busca la función de entre un espacio de funciones, que ajusta los datos de entrada con los de salida, en este caso, los datos de precipitación y caudal con los de caudal. Es obvio que la disponibilidad de mayor información de la cuenca sería deseable, pero con la información de precipitación u caudal disponibles se han logrado buenos resultados.

PROLOGO.

En Hidrología e Hidráulica existen básicamente tres tipos de modelos para investigar los fenómenos que son objeto de su estudio. Primero, los modelos físicos o a escala que necesitan de mucha información y sus costos son elevados, además muchas veces los fenómenos analizados no son replicables en laboratorio. Segundo, los modelos matemáticos, que son basados en el comportamiento del fenómeno o en primeros principios, por esto se necesita mucha información y la calibración del modelo requiere de muchos parámetros físicos, que usualmente son difíciles de obtener y la mayoría de las veces costosos. Finalmente los modelos empíricos, en los que están involucradas ecuaciones matemáticas pero que provienen del procesamiento de datos de entrada y salida en series de tiempo. Típicos ejemplos de estos son el método del hidrograma unitario, modelos estadísticos, y en la última década los llamados data-driven models o modelos basados en los datos.

En esta tesis se analiza la robustez del método de las redes neuronales para la predicción de caudales aplicado a la cuenca alta del Rio Paute. El método de las redes neuronales es un modelo basado en los datos, por lo tanto es un modelo empírico, pero que la literatura respalda por su alta calidad de resultados en predicción, generalización y facilidad de implementación.

En esta tesis se ha verificado las capacidades anteriormente descritas para la aplicación de Redes Neuronales, además se ha desarrollado una metodología para evaluar su robustez, y finalmente se ha implementado un sistema con capacidad de respuesta en tiempo real para el problema de predicción de caudales.

AGRADECIMIENTOS.

Esta investigación se hizo posible gracias al financiamiento del Grupo de la Tierra y del Ambiente, de la Universidad de Cuenca. Datos, infraestructura y materiales fueron facilitados por diversas instituciones mediante la gestión del Grupo de la Tierra y del Ambiente. Por esto agradezco profundamente el apoyo y la confianza que sus directivos y personal me han brindado.

Deseo expresar mi sincero agradecimiento a mi director de tesis Esteban Samaniego PhD. MSc. Ing. que con sus conocimientos y fundamentados criterios guiaron mi trabajo a lo largo de toda su realización.

Mi gratitud por el tiempo y las observaciones que siempre me alentaron hacia nuevos rumbos de esta investigación al Dr. Piercosimo Tripaldi.

Agradezco el apoyo, paciencia y compresión que mi familia me ha brindado durante la realización de esta maestría.

Lenin Vladimir Campozano Parra

Cuenca, Abril 2011

PARTE 1

INTRODUCCIÓN Y DESARROLLO TEÓRICO

CAPITULO 1

INTRODUCCIÓN

CAPÍTULO 1

INTRODUCCIÓN

1.1Generalidades

En Hidrología e Hidráulica existen básicamente tres tipos de modelos para investigar los fenómenos que son objeto de su estudio. Primero, los modelos físicos o a escala que necesitan de mucha información y sus costos son elevados, además muchas veces los fenómenos analizados no son replicables en laboratorio. Segundo, los modelos matemáticos, que son basados en el comportamiento del fenómeno o en primeros principios, por esto se necesita mucha información y la calibración del modelo es necesario para determinar valores de parámetros físicos, lo cual usualmente no es sencillo. Finalmente los modelos empíricos, en los que están involucradas ecuaciones matemáticas pero que provienen del procesamiento de datos de entrada y salida en series de tiempo. Típicos ejemplos de estos son el método del hidrograma unitario, modelos estadísticos, y en la última década los llamados data-driven models o modelos basados en los datos.

En esta tesis se analiza la robustez del método de las redes neuronales para la predicción de caudales aplicado a la cuenca alta del Rio Paute. El método de las redes neuronales es un modelo basado en los datos o data-driven model. El análisis de la robustez del modelo nos sirve para evaluar las variaciones en la respuesta del modelo debido a variaciones en los datos de entrada. Las variaciones de los datos de entrada puede ser debido a la variabilidad propia de las variables de entrada (suponiendo que no existe errores en los mismos), por un lado, y pueden deberse también a errores en lectura, transcripción, calculo, etc. Por esta razón se realiza un análisis de sensibilidad y un análisis de incertidumbre para estudiar la robustez del modelo. Si bien es cierto que durante las dos últimas décadas se ha investigado sobre estos dos tópicos, no existe una definición científica única de ellas, y su procedimiento depende del campo de aplicación.

Del análisis de sensibilidad se busca obtener información sobre la importancia de cada variable en el modelo, así como los intervalos de respuesta del modelo frente a las variaciones en las variables de entrada más importantes. Del análisis de incertidumbre se busca obtener información sobre el nivel de incertidumbre del modelo, específicamente para el problema de predicción de caudales. Es decir, los errores entre

datos observados y calculados, para un grupo de datos que no hayan intervenido en la calibración del modelo, serán utilizados para calcular la incertidumbre en predicción.

La predicción de caudales en un punto dado de una cuenca hidrográfica es de especial importancia para la operación de estructuras hidráulicas, para la operación de diversos sistemas, para la generación de energía eléctrica, para la toma de decisiones en los más diversos tópicos del manejo de recursos hídricos. Para dicha predicción se ha utilizado modelos matemáticos, estando los mismos basados en procesos hidrológicos. Sin embargo con la evolución de las capacidades computacionales, se han desarrollado modelos de predicción de caudales fundamentados en el análisis de datos o data-driven models. Estos métodos están basados en inteligencia computacional y aprendizaje de máquina.

El método de las redes neuronales será utilizado para la predicción de caudales, dados los datos de precipitación y caudal como variables de entrada. Es importante recalcar que los datos se han obtenido por mediación del Grupo de la Tierra y del Ambiente [2], ante diversas entidades que han monitoreado la cuenca del Rio Paute. Las estaciones de registro de precipitación son: Cancan-Soldados, Chanlud, El Portete, El Labrado, Machangara-DJ Chulco, Matadero en Sayausí, Tarqui - DJ.Cumbe, Ucubamba_en_ETAPA, Yanuncay_en_Pucan y la estación de registro de caudal es en el Tomebamba_en Ucubamba. Debido a que el modelo de redes neuronales recibe la información de cada una de las estaciones, sin que los datos de precipitación hayan sido concentrados, entonces el modelo estudiado es distribuido y es un modelo basado en los datos. Contrariamente, para la elaboración de un modelo matemático distribuido es necesario información detallada de la cuenca, como tipo de suelo, pendientes del terreno, vegetación, uso del suelo, evaporación, evapotranspiración, etc, lo cual hace complicado su implementación y operación en tiempo real. El uso de redes neuronales para la elaboración de un modelo distribuido tiene la ventaja de la no dependencia de estos factores debido a que la red busca la función de entre un espacio de funciones, que ajusta los datos de entrada con los de salida, en este caso, los datos de precipitación y caudal con los de caudal. Es obvio que la disponibilidad de mayor información de la cuenca sería deseable, pero con la información de precipitación disponible se han logrado buenos resultados.

La arquitectura estándar de una red neuronal óptima para un problema de predicción no está definida, debido a que la complejidad de la misma depende del problema analizado. Entonces la red adecuada para un grupo de datos establecidos se puede obtener mediante la optimización de un parámetro estadístico con respecto al número de capas, número de neuronas, y tipos de función de activación. Por esta razón se ha decidido utilizar el Toolbox de Matlab, con el cual se puede implementar arquitecturas de red, de manera versátil, confiable, y dentro de un ambiente amigable al usuario. El uso del Toolbox de redes neuronales ha acelerado el proceso de selección de la red más adecuada para los datos y el problema planteado. Queda como trabajo posterior la

programación de una red neuronal específica para esta tarea, una vez que mediante el toolbox de Matlab se ha logrado determinar la red óptima.

1.2 Motivación

Tres son las razones principales que hacen importante la investigación que se desarrolla en esta tesis.

Como primera motivación del tema es que desde el punto de vista netamente científico es relevante la validación de modelos frente a la realidad física. Como se ha mencionado anteriormente, en esta tesis el modelo a ser validado es el de redes neuronales para la predicción de caudales en un punto de la cuenca alta del rio Paute, con un tiempo de avance de un día (lead time).

La segunda es netamente funcional, ya que el tener una herramienta computacional confiable en tiempo real para predicción de caudales con tiempo de avance de un día es muy importante para operaciones y maniobras dentro del proyecto hidroeléctrico Paute.

La tercera es incursionar en el uso de modelos data-driven, por razones de facilidad de implementación, ya que no se necesita la información requerida por los modelos físicos o matemáticos, y por su costo relativamente inferior al de los modelos citados anteriormente.

1.3 Objetivos

1.3.1 Objetivo general

Desarrollar el análisis de la robustez de un método de predicción de caudales mediante Redes Neuronales aplicado a la cuenca del rio Paute.

1.3.2 Objetivos específicos

- Encontrar una red neuronal óptima para el problema de predicción de caudales con un día de avance, en la cuenca alta de río Paute.

- Proponer una definición de robustez del modelo de redes neuronales.

- Aplicar la definición de robustez propuesta y validarla frente al modelo estudiado.

1.4 Esquema de tesis

Por motivos de organización y coherencia interna esta tesis se desarrolla en dos partes.

La Parte 1: INTRODUCCIÓN Y DESARROLLO TEÓRICO, está destinada al acercamiento y ubicación temática; además se exponen los conocimientos teóricos necesarios para su desarrollo y aplicación posterior al problema. Esta primera parte está compuesta de tres capítulos.

El capítulo 1: INTRODUCCIÓN. En este capítulo se expone generalidades del tema, motivación, objetivos generales y específicos, y finalmente el esquema de tesis.

El capítulo 2: REDES NEURONALES ARTIFICIALES. Se presenta el desarrollo teórico de las redes neuronales. Origen del modelo, arquitecturas de red y algoritmos de aprendizaje.

El capítulo 3: DEFINICIÓN DE LA ROBUSTEZ DE UN MODELO, debido a que no existe una definición generalizada del concepto de robustez, más bien esta depende del campo de aplicación, en este capítulo definimos la robustez para este modelo, y posteriormente se sustenta teóricamente el análisis de sensibilidad y el análisis de incertidumbre.

La Parte 2: IMPLEMENTACIÓN DE UN MODELO DE REDES NEURONALES Y ANÁLISIS DE ROBUSTEZ, está orientada a la aplicación específica de los desarrollos teóricos expuestos en la Parte 1 a la situación problémica y los datos disponibles. En la Parte 2 es necesario exponer en primera instancia la implementación varias redes neuronales, luego determinar el modelo óptimo y finalmente procedemos al análisis de robustez de la o las redes que mejor resuelvan el problema de predicción de caudales. Esta segunda parte está compuesta de tres capítulos.

El capítulo 4: PRE PROCESAMIENTO DE LOS DATOS. El preprocesamiento de datos de precipitación y caudal medidos en las estaciones de la cuenca alta del río

Paute, pueden procesarse previamente a su uso en las redes neuronales, debido a dos razones.

La primera, debido a que existen relaciones estadísticas o dependencia en mayor o menor grado entre estaciones, es importante identificar este nivel de dependencia, para luego alimentar el modelo con información que no sea redundante. Para este análisis se han aplicado las funciones de correlación cruzada, auto-correlación y análisis de componentes principales.

La segunda razón para hacer un pre procesamiento de datos está relacionada con la factibilidad de procesamiento de la información por parte de la red. Como veremos en desarrollos teóricos posteriores, las redes neuronales emplean las llamadas funciones de activación en cada neurona. Las funciones utilizadas en nuestro estudio son las llamadas sigmoidales, las cuales tienen como dominio (-∞; ∞) y mapean este dominio al rango [0; 1]. Por esta razón se realiza un proceso de escalado de datos para que la red pueda ajustarse a los datos.

El capítulo 5: IMPLEMENTACIÓN DE ARQUITECTURAS DE REDES NEURONALES Y RED ÓPTIMA. Este capítulo está orientado a la implementación de varias arquitecturas de red utilizando el toolbox de Matlab, orientadas a la predicción de caudales con un día de avance. La eficiencia de los resultados de estas redes es presentada como el valor de Q^2 para predicción. Posteriormente se procede a encontrar la red óptima, para finalmente hacer un análisis de los resultados de la red óptima para caudales bajos, medios y altos, que serán definidos en la sección correspondiente.

El capítulo 6: ROBUSTEZ DEL MODELO PROPUESTO. Con el objeto de analizar la robustez del modelo previamente definido en el capítulo 3, se realiza en este capítulo el análisis de sensibilidad del modelo, con el objeto de identificar las estaciones a las cuales es más sensible el modelo. Posteriormente se desarrolla la parte complementaria del análisis de sensibilidad que es el análisis de incertidumbre, en el cual se calcula la incertidumbre debido a incertidumbre estructural del modelo y debido a incertidumbre en las mediciones de precipitación y la posterior propagación atraves del modelo.

El capítulo 7: CONCLUSIONES Y TRABAJO A FUTURO. En este capítulo final se recopila los resultados obtenidos en los capítulos 4, 5, y 6 y se hace un análisis y conclusiones del trabajo realizado. Debido a que este trabajo ha sido de carácter altamente investigativo se han identificado varios temas que necesitan mayor estudio, los cuales sobrepasan los objetivos de esta tesis pero que su desarrollo sería de mucha importancia para la mejor compresión del tema tratado. Estos temas son expuestos en la sección de Trabajo a Futuro.

CAPITULO 2

REDES NEURONALES ARTIFICIALES

CAPÍTULO 2

REDES NEURONALES ARTIFICIALES

2.1 Introducción

Un modelo simplificado de una neurona real está compuesto del cuerpo, un conjunto de fibras que llegan del cuerpo llamadas dendritas, y una fibra que sale del cuerpo llamada axón. 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 otra depende de la eficiencia de la transmisión sináptica. Esta eficiencia podemos llamarle variable w. si la eficiencia es positiva entonces la sinapsis es llamada excitatória, si es negativa es llamada inhibitoria. Una neurona real puede variar la cantidad de impulsos por segundo, dependiendo de su actividad. Para modelar esta actividad se ha simplificado este comportamiento realizando modelos tipo binario, es decir, transmisión o no transmisión.(Ver figura 1)

El modelo simplificado de una neurona puede ser simulado por una neurona artificial.

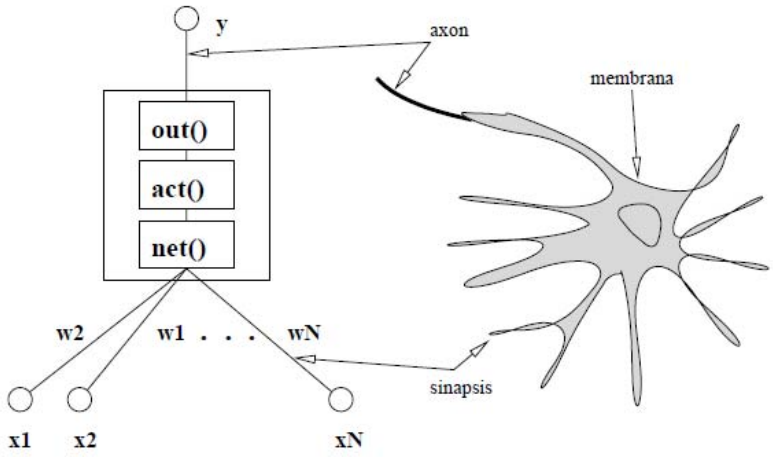

Figura 1 *Modelo simplificado de una neurona*

La transmisión de información mediante las dendritas que dependen del tiempo está representado por las variables x. la eficiencia de la transmisión está representado por los pesos w. El procesamiento del cuerpo de la neurona se procesa por funciones internas que generan la magnitud de salida out. El axón se modela mediante la variable de salida y.

2.2 Neuronas con argumento escalar

Vamos a describir el funcionamiento de una neurona sin umbral (término bias). Una neurona con un valor escalar p como ingreso, transmitido mediante la conexión de peso w tiene el valor wp. Este valor llega a la neurona y es el argumento de la función de transferencia f, la cual produce el escalar $a = f(wp)$. Para simplificar nomenclatura podemos definir como n, el escalar que ingresa a la neurona, y que será el argumento de f, es decir $a = f(n)$. El funcionamiento de una neurona con umbral es el siguiente. El ingreso mediante el peso w tiene el valor wp como en el caso anterior. El valor de ingreso en el umbral es 1, mientras que su peso es b. por lo tanto b y wp se suman para ser argumento de f, la función de transferencia. La función del umbral es como la de un peso b que se suma a wp [1].

Es importante observar que $a = f(wp + b)$ que es la respuesta de la neurona con umbral, pretende mapear la entrada p mediante los parámetros w y b. El objetivo de esta neurona es entonces mapear p a un a* deseado mediante el ajuste de w y b. la diferencia entre a y a* se deberá ir disminuyendo hasta un valor aceptable. Para que los parámetros w y p se ajusten será necesario un algoritmo de entrenamiento. Más adelante en el desarrollo de esta tesis nos enfocaremos en los algoritmos de aprendizaje más utilizados y aplicables a la temática estudiada. En la fig.2 representamos el comportamiento de las neuronas tratadas.

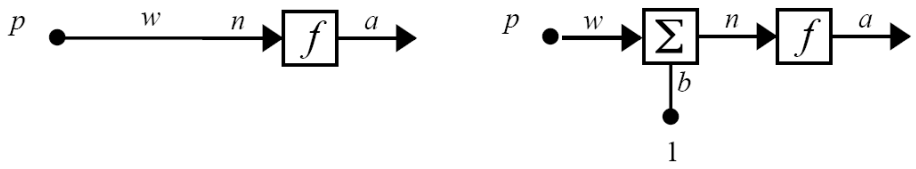

Figura 2 *Modelo de una Neurona.*

Izquierda neurona sin umbral. Derecha neurona con umbral

2.3 Funciones de transferencia

Existen diferentes tipos de funciones de transferencia [1], cuyo argumento puede o no incluir el término umbral. Las funciones de transferencia son también conocidas como funciones de activación. Las más utilizadas serán descritas a continuación.

2.3.1 Función Escalón o limitador fuerte

Esta función de transferencia limita la respuesta de la neurona a 0 cuando el argumento n es menor que 0, o a 1 cuando el argumento n es mayor que 0. Recordemos que el argumento n puede contener o no el término umbral, según como se desea programar la neurona. Ver fig. 3.

$$a = 0 \quad si\, n < 0$$

$$a = 1 \quad si\, n > 0$$

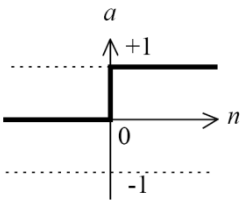

Figura 3 *Función de transferencia escalón*

2.3.2 Función lineal

Esta función de transferencia es utilizada para una aproximación lineal. Ver fig. 4. Esta función de transferencia tiene por argumento n y si su salida es “a”, entonces se define como:

$$a = b + c * n$$

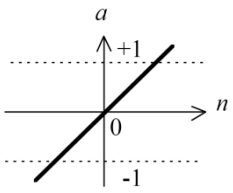

Figura 4 *Función de transferencia lineal*

2.3.3 Función sigmoidal

Esta función de transferencia es muy utilizada, particularmente en problemas de ajuste de datos. Ver fig. 5. Esta función de transferencia toma el argumento n y lo mapea al rango [0,1], según la siguiente ecuación: $a = \frac{1}{1+\exp(-n)}$

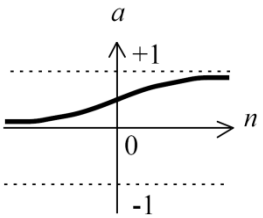

Figura 5 *Función de transferencia sigmoidal*

2.3.4 Resumen de funciones de transferencia

A continuación presentamos en la Tabla. 1 un resumen de varias funciones de transferencia que pueden ser utilizadas para diversas aplicaciones.

Recordamos que según la nomenclatura utilizada el valor del argumento es n, mientras que la salida o repuesta de la red es a.

Nombre	Respuesta
Limitador Fuerte	$a = 0 \;\; n < 0; \;\; a = 1 \;\; n \geq 0$
Limitador Fuerte Simétrico	$a = -1 \;\; n < 0; \;\; a = 1 \;\; n \geq$
Lineal Positiva	$a = 0 \;\; n < 0; \;\; a = n \;\; n \geq 0$
Lineal	$a = n$
Lineal Saturado	$a = 0 \;\; n < 0; \; a = n \;\; 0 \leq n \leq 1; a = 1 \;\; n > 1$
Lineal Saturado Simétrico	$a = -1 \;\; n < -1; \; a = n \;\; -1 \leq n \leq 1$ $a = 1 \;\; n > 1$
Sigmoidal Logarítmico	$a = \frac{1}{1 + \exp(-n)}$
Tangente Sigmoidal Hiperbólica	$a = \frac{\exp(n) - \exp(-n)}{\exp(n) + \exp(-n)}$
Competitiva	$a = 1 \; neurona \; con \; máximo \; argumento \; n$ $a = 0 \; neuronas \; con \; argumento \; menor$

Tabla 1 *Resumen de funciones de transferencia*

2.4 Neuronas con argumento vectorial

Sea $p = [p_1; p_2; \ldots\ldots; p_R]$ el vector de entrada a una neurona i, entonces cada componente del vector será multiplicado por un peso del vector $W = [w_{(i,1)}; w_{(i,2)}; \ldots\ldots; w_{(i,R)}]$ resultando el argumento $Wp = [p_1 w_{(i,1)}; p_2 w_{(i,2)}; \ldots\ldots; p_R w_{(i,R)}]$ que puede o no ser sumado al término umbral de la neurona i, b_i. Entonces la respuesta a_i de la neurona con función de transferencia f es:

$$a_i = f(n_i) = f(W_{(i,j)} p_j + b_i)$$

$$n_i = W_{(i,j)} p_j + b_i$$

En la fig. 6 podemos simbolizar el funcionamiento de una neurona con argumento vectorial.

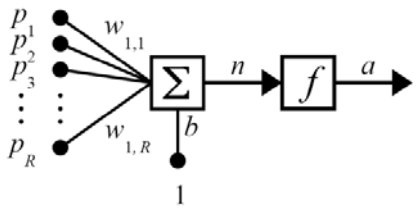

Figura 6 *Neurona con argumento vectorial*

2.5 Topología de una red neuronal

En una red neuronal múltiples neuronas son interconectadas para formar una red, con el objeto de potenciar la capacidad de cómputo [5]. La configuración de las interconexiones puede ser descrita con un grafo dirigido (fig. 7). Un grafo dirigido está compuesto de nodos y de arcos dirigidos. La topología del grafo puede ser categorizado como cíclico o acíclico.

Una red acíclica, consiste en arcos sin regreso, es decir el flujo de información se dirige desde el inicio hasta el final de la red, sin tener ningún retorno, es por eso que estas redes también se denominan feedforward neural networks. Redes de este tipo se utilizan para realizar mapeos no lineales entre los datos de entrada y los datos de salida. Contrariamente en una red cíclica, los arcos forman retornos hacia neuronas anteriores, este tipo de redes también se denominas recurrentes. Debido a los bucles recurrentes de información estas redes son dinámicas no lineales y además poseen memoria interna. Las redes neuronales recurrentes usualmente presentan comportamientos muy complejos, es por esto que todavía son objeto de investigación.

Este tipo de redes también son llamadas feedback neural networks.

Figura 7 *Topologías de red.*

Izquierda Red Acíclica. Derecha red cíclica

2.6 Arquitecturas de red

La clasificación de las redes neuronales en clase feedback y clase feedforward es bastante general. Por lo tanto vamos a estudiar varias arquitecturas ampliamente utilizadas para aplicaciones comunes. Muchas otras arquitecturas que se encuentran en la bibliografía especializada no son estudiadas debido a que las arquitecturas adoptadas dependen del problema estudiado.

En principio varias neuronas pueden recibir la información de entrada, y estas a su vez enviar información a una neurona de salida que es la respuesta de la red. Este arreglo de neuronas que estaría trabajando en forma interconectada llamamos red neuronal. También es posible organizar arreglos de varias capas que al final dan una respuesta del arreglo neuronal. Este arreglo también es una red neuronal, pero más compleja.

Generalmente las redes neuronales tienen un arreglo de capas tal que podemos identificar claramente la capa de entrada, una o varias capas ocultas, y finalmente la capa de salida. La capa de entrada no altera la información entrante, tan solo se encarga de repartir la información a las neuronas de la primera capa oculta. La o las capas ocultas tienen interconexión para procesar la información, hasta que estas envían la información a la capa de salida que es la respuesta del sistema. El término capa oculta se debe a que el programador o usuario envía los datos a la capa entrante y recibe datos de la capa saliente, todo el procesamiento interno de las capas ocultas no es muy evidente como para dar un significado. (Ver Fig. 8)

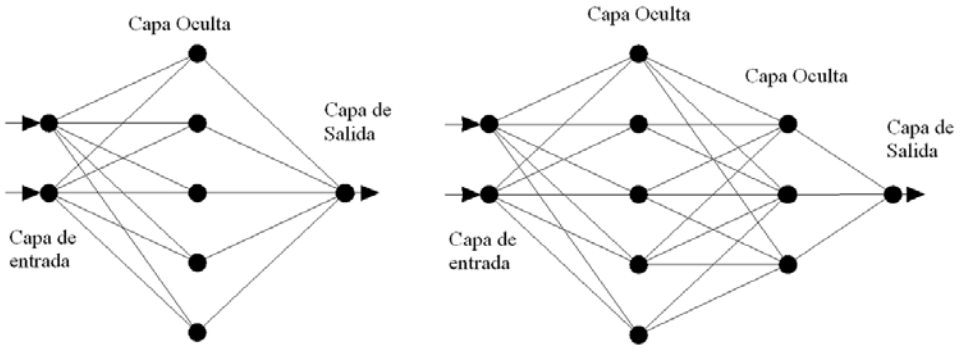

Figura 8 *Redes multicapas.*

Izq: Una capa oculta. Der: Dos capas ocultas.

2.6.1 Multilayer Perceptron

El modelo llamado Multilayer Perceptron (MLP), consiste en una red tipo feedforward, la cual tiene varias capas de neuronas. Cada neurona en un MLP tiene una función de activación que es usualmente no lineal, continua y diferenciable. Son bastante usadas la función sigmoidal y la tangente hiperbólica [5].

La configuración de una red tipo MLP está compuesta por una capa de entrada, la cual no tiene funciones de neurona, es decir, la información entrante no es alterada. Luego siguen las capas ocultas, y finalmente la capa de salida. Cualquier configuración que sea aciclica es básicamente un MLP.

Un MLP genera un mapeo no lineal entre los datos de entrada y los de salida. El siguiente ejemplo nos ayuda a mostrar las potencialidades de esta arquitectura de red neuronal.

Sean los datos de entrada los pares ordenados (x, y) dentro del dominio bidimensional $0 \leq x \leq 1; 0 \leq y \leq 1$; estos pares ordenados ingresan como datos a una red MLP con función de activación sigmoidal y mapea a la respuesta z que se muestra en la figura 9. Ha sido probado que un MLP con suficientes neuronas y un máximo de dos capas ocultas puede generar un mapeo complejo dentro de un rango finito [6].

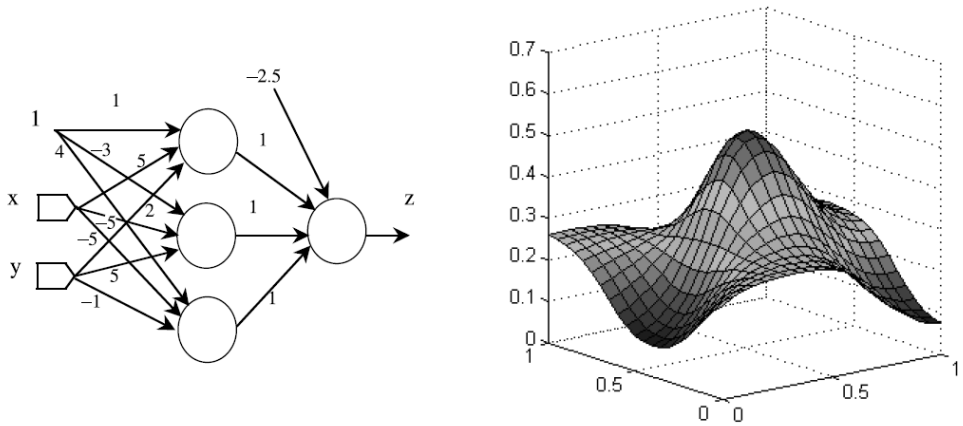

Figura 9 *Mapeo de un dominio bidimensional mediante un MLP*

2.6.2 Radial Basis Networks (RBN)

La redes llamadas Radial Basis Network (RBN) es una red tipo feedforward, cuyas neuronas tienen como función de activación las funciones llamadas Radial Basis [5]. Estas funciones son de la forma $f(\|x - m_0\|) = f(r)$. Esta función es simetrica respecto a un punto central m_0. Presentamos en la figura 10 algunos ejemplos de funciones base radial en una dimensión.

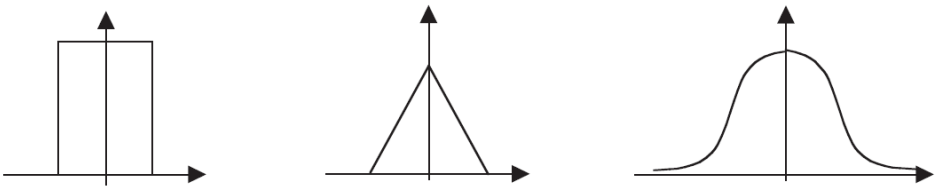

Figura 10 *Funciones de activación tipo Radial Basis. Tres ejemplos.*

Estas funciones pueden ser utilizadas para aproximar cualquier función mediante su combinación lineal (Ver Fig. 11). Esta superposición puede entenderse como la superposición de funciones sinusoidales y cosenoidales en las series de Fourier.

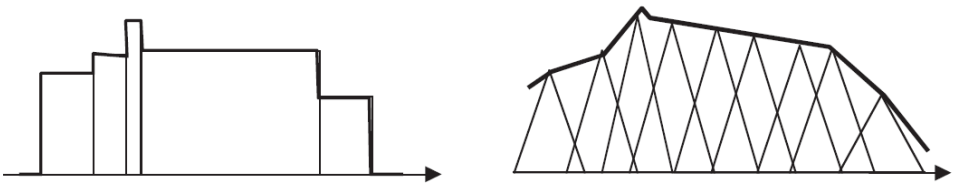

Figura 11 *Aproximación de una función mediante la combinación lineal de funciones radiales*

La función aproximada se representa como una combinación lineal ponderada de una familia de funciones base radiales con diferentes escalas y traslaciones.

$$\hat{F}(x) = \sum_{i=1}^{C} w_i \varphi(\|x - m_i\|/\sigma_i)$$

En el planteamiento anterior w_i son los pesos o ponderaciones de las funciones $\varphi(\|x - m_i\|/\sigma_i)$.

2.6.2.1 Formulación del problema de aproximación de funciones base radiales

Dado un conjunto de puntos $\{x(k); 1 \leq k \leq K\}$ y los valores de una función $F(x)$ desconocida evaluada en esos K puntos $\{d(k) = F(x(k)); 1 \leq k \leq K\}$, encontrar una aproximación de $F(x)$ en la forma $\hat{F}(x) = \sum_{i=1}^{C} w_i \varphi(\|x - m_i\|/\sigma_i)$ tal que la suma del cuadrado del error de aproximación en esos puntos de entrenamiento

$$\sum_{k=1}^{K} [d(k) - \hat{F}(x(k))]^2$$

es minimizada.

Según el planteamiento para la solución del problema anterior las RBN pueden clasificarse como tipo I y tipo II.

RBN Tipo I

El primer tipo de RBN elige cada muestra de entrenamiento como la ubicación de una función base radial. Por lo tanto C=K y $m_i = x(i)$, donde $1 \leq i \leq K$. Además un parámetro σ es elegido para cada función. Por conveniencia $\sigma_i = 1$. Escribiendo la formulación en forma matricial:

$$\begin{bmatrix} \varphi(\|x(1) - m_1\|) & \varphi(\|x(1) - m_2\|) & \varphi(\|x(1) - m_C\|) \\ \varphi(\|x(2) - m_1\|) & \varphi(\|x(2) - m_2\|) & \varphi(\|x(2) - m_C\|) \\ \dots\dots\dots\dots\dots\dots & \dots\dots\dots\dots\dots\dots\dots & \dots\dots\dots\dots\dots \\ \varphi(\|x(K) - m_1\|) & \varphi(\|x(K) - m_2\|) & \varphi(\|x(K) - m_C\|) \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \dots. \\ w_C \end{bmatrix} = \begin{bmatrix} d(1) \\ d(2) \\ \dots. \\ d(K) \end{bmatrix}$$

Llamemos $\mathbf{\Phi}$ la matriz de KxC elementos, en la cual K=C, y normalmente es definida positiva para funciones bases radiales comúnmente usadas, por lo tanto el vector de pesos W puede ser determinado mediante:

$$\mathbf{W} = \mathbf{\Phi}^{-1}\mathbf{d}$$

Sin embargo en aplicaciones prácticas esta matriz $\mathbf{\Phi}$ es cercana a singular llevando a inestabilidades numéricas en su solución. Diferentes métodos pueden ser aplicados para solucionar este problema.

El primer método es el de regularización. Sea $\boldsymbol{\lambda}$ un numero positivo pequeño, entonces:

$$\mathbf{W} = (\mathbf{\Phi} + \boldsymbol{\lambda}\mathbf{I})^{-1}\mathbf{d}$$

El segundo método es mínimos cuadrados. El objetivo es hallar la solución W_{LS} tal que $\|\mathbf{\Phi W} - \mathbf{d}\|^2$ es minimizada. Aquí:

$$\mathbf{W} = (\mathbf{\Phi})^{+}\mathbf{d}$$

Donde $(\mathbf{\Phi})^{+}$es la matriz seudo-inversa., que puede ser encontrada usando la descomposición de valores singulares.

RBN Tipo II

El tipo II está basado en la teoría de regularización [7]. La función base radial utilizada es la Gausiana.

$$\varphi(\,\|x - m\|) = exp(-\frac{\|x - m\|^2}{2\sigma^2})$$

El problema es minimizar $\|GW - d\|^2$ sujeto a la restricción $W^T G_0 W = a$ constante. En este planteamiento se ha utilizado G como $\mathbf{\Phi}$para especificar el uso de funciones Gausianas.

La solución al problema de optimización puede ser encontrado como:

$$W = (G^T G + \lambda G_0)^{-1} G^T d$$

Donde λ es un parámetro de optimización y usualmente es elegido como un número pequeño no negativo. Como λ tiende a cero esta solución se convierte a la solución de mínimos cuadrados.

2.6.3 Redes de Aprendizaje Competitivo (CLN)

En los dos tipos de redes vistas, MLP y RBF, el aprendizaje se basa en la reducción paulatina del error entre datos observados y calculados, esto significa que la adaptación de los pesos sinápticos está guiada por la reducción de este error. En las redes que vamos a estudiar en esta sección el criterio a aplicar es aprendizaje competitivo [5].

Con las redes tipo CLN o Competitive Learning Networks, las neuronas de una misma capa compiten entre ellas para representar el vector de entrada. La neurona que mejor representa el vector de entrada, ajustará su peso sináptico. Como tal, el aprendizaje competitivo puede ser considerado como un algoritmo de clustering secuencial.

2.6.3.1 Redes ortogonales lineales

En una red lineal de una capa, la respuesta de la capa n es (Fig. 12):

$$y_n(t) = \sum_{m=1}^{M} w_{nm}(t) x_m(t)$$

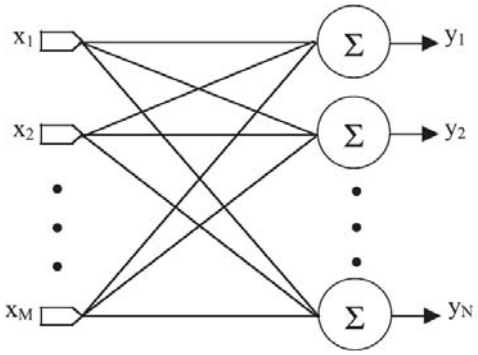

Figura 12 *Red de mapeo lineal* $\boldsymbol{R^M \rightarrow R^N}$

Los pesos sinápticos son actualizados según una regla Hebbian generalizada mediante:

$$\Delta w_{nm}(t) = w_{nm}(t+1) - w_{nm}(t) = \eta y_n(t)[x_m(t) - \sum_{k=1}^{n} w_{km}(t) y_k(t)]$$

El vector de pesos sinápticos w_n convergerá al vector propio del n-ésimo valor propio más grande de la matriz de covarianza de la matriz de observaciones formada por:

$$C = \sum_{t} X(t) X^T(t)$$

Es decir que luego de la convergencia esta red producirá las componentes principales (vectores propios) de la matriz de covarianza de los vectores de entrada o ingreso. El análisis de componentes principales (PCA) ha encontrado aplicaciones diversas siendo esta una de ellas.

2.6.3.2 Mapeos Self-Organizing

Una red Self-Organizing Map SOM, es una red neuronal competitiva de una capa, que impone un orden pre asignado entre neuronas [7]. Esta red es entrenada mediante un aprendizaje no supervisado, para producir una representación discretizada (usualmente de dos dimensiones), del espacio de vectores de ingreso. Esta representación se llama mapeo. La diferencia respecto a otras redes neuronales es que estas utilizan una función de las neuronas vecinas para preservar las propiedades topológicas del espacio de entrada. Esto hace las SOM útiles para visualizar datos de alta dimensionalidad. Estas redes son llamadas también Kohonen map. Las SOM usualmente trabajan con dos grupos de datos, los de entrenamiento y los de mapeo.

Los de entrenamiento construyen la red utilizando los vectores de ingreso mediante un proceso llamado cuantización vectorial (debido a que el proceso clasifica los vectores de ingreso en grupos, tiene sentido esta denominación ya que los resultados se presentan en grupos de datos). Luego de entrenada la red, se procede a la clasificación o agrupamiento con datos nuevos, este proceso se llama mapeo.

La configuración de la capa considerada puede extenderse a otras topologías [9]. La motivación para utilizar otras topologías es el eliminar los efectos de los bordes. Una aplicación de TSOFM (Toroidal Self Organizing Map) es en reconocimiento de imágenes, en donde se ha eliminado los efectos de borde y obviamente se ha incrementado la tasa de reconocimiento (Fig. 13).

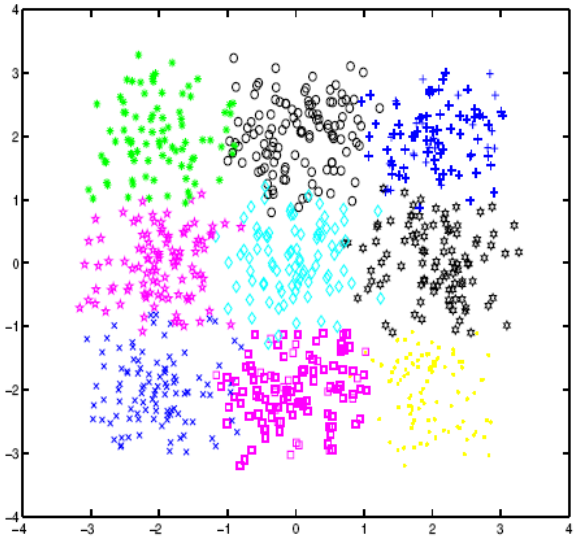

Figura 13 *Representación de datos agrupados mediante una red Self-Organizing-Map.*

Como se puede ver la red ha clasificado los datos en nueve grupos

2.6.4 Committee Machines

Una red tipo committee machines está compuesta de varias redes neuronales [7]. Los vectores de ingreso de datos acceden a cada red, de cada red o módulo se obtiene una respuesta, la cual se transmite al modulo final, el cual recibe la respuesta de todos los módulos y realiza una decisión basada en la respuesta colectiva.

El módulo de decisión puede generar la respuesta final de la red committee machines según dos criterios.

El primer criterio es ensamblaje. En este tipo de redes la decisión de respuesta combina por mayoría o por una ponderación de respuestas, pero esta respuesta es independiente de los datos de entrada. Por esta razón estas redes son llamadas estáticas.

El segundo criterio es relacionar la respuesta del modulo final en función de los datos de entrada. Por esta razón este tipo de redes son llamadas dinámicas. Son características las redes llamadas Mixture of Expert (MOE). Estas redes fraccionan el espacio de los vectores de entrada y especializan cada módulo en una subregión del problema. Una red de entrada asignara datos a ciertos módulos, estos módulos procesan estos datos y envían la respuesta. Esto significa que el funcionamiento de este Committee Machines depende de los datos de entrada.

2.7 Aprendizaje

Es necesario definir el significado de aprendizaje relacionado con inteligencia artificial. Aprendizaje significa que dado un conjunto de observaciones y una estructura de función se utiliza un algoritmo para encontrar una función específica que optimiza alguna función de costo relacionada con las observaciones.

Usualmente es posible definir una función de costo por diversas características de la misma. Pero básicamente la función de costo depende de la tarea de aprendizaje necesario.

Las tareas de aprendizaje son primordialmente tres [24].

Aprendizaje supervisado: en este tipo de aprendizaje dados datos de entrada X y de salida Y busca encontrar la función que mapea X a Y. La función de costo en este tipo de aprendizaje generalmente está relacionada con el error entre la predicción de la función y los datos reales.

Usualmente es utilizado el error cuadrático medio como función de costo. Utilizando esta función de costo con el algoritmo gradient- descent para optimizarlo aplicado a redes neuronales, este algoritmo es llamado backpropagation.

Los problemas que son usualmente tratados con aprendizaje supervisado son clasificación y regresión.

Aprendizaje no supervisado: en este tipo de aprendizaje unos datos X son dados y la función de costo a ser minimizada puede ser función de los datos X y de la respuesta de la red F. La función de costo es dependiente de la tarea a realizarse. Contrariamente a funciones de costo utilizadas en aprendizaje supervisado, las funciones de costo podrían ser maximizadas dependiendo del problema.

Tareas que utilizan este tipo de aprendizaje son clustering, compresión, filtrado y problemas de estimación en general.

Aprendizaje Reforzado: en este tipo de aprendizaje los datos X (t) generalmente no son dados, más bien estos son generados por la interacción de un agente con el ambiente. En cada tiempo t, el agente actúa mediante una variable Y (t) y el ambiente genera una observación X (t) y una función de costo C (t) de acuerdo a cierta dinámica. El objetivo es determinar lineamientos para identificar las acciones que minimizan el costo a largo plazo.

El ambiente es usualmente simulado mediante Procesos de Decisión de Markov. Los estados S_i y las acciones A_i y las distribuciones de probabilidad de costo $P(C_i|S_i)$ de observación $P(X_i|S_i)$ y de transición $P(S_{i+1}|S_i\,,A_i)$, mientras que los lineamientos son definidos distribuciones condicionales sobre las acciones dadas las observaciones.

Los problemas típicos a ser resueltos por este tipo de aprendizaje son problemas de control.

En las secciones siguientes estudiamos algunos algoritmos de aprendizaje.

2.7.1 Algoritmo Levenberg-Marquardt

El algoritmo de Levenberg-Marquardt es una técnica iterativa que localiza el mínimo de una función multivariada que es expresada como la suma de los cuadrados de una función real no lineal [11]. Esta técnica se ha utilizado ampliamente en diversos

campos de aplicación. Cuando la solución actual es distante de la correcta, el algoritmo actúa como un método steepest descent, es decir, es lento pero asegura convergencia. Por lo contrario cuando la solución está cerca de la correcta, este algoritmo se convierte en el algoritmo de Gauss-Newton. A continuación se hará una descripción del método.

Sea f una relación funcional la cual mapea un vector de parámetros $p \in R^n$ a un vector estimado de medición $\hat{x} = f(p)$, $\hat{x} \in R^n$. Una estimación inicial del vector de parámetros es p_o y un vector de medición x son conocidos, y se desea encontrar un vector p^+ que satisface la relación funcional f. Básicamente el algoritmo aproxima linealmente f en el vecindario de p. Para variaciones pequeñas de $\|\delta_p\|$, una aproximación en serie de Taylor nos lleva a:

$$f(p + \delta_p) \approx f(p) + J\delta_p$$

En donde J es el jacoviano de la matriz $\frac{\partial f(p)}{\partial p}$.

El proceso iterativo empieza con p_o, el método produce una serie de vectores $p_1, p_2, \dots, p_q$ que converge hacia el vector minimizador p^+ . En cada iteración se encuentra δ_p tal que minimiza $\|x - f(p + \delta_p)\| \approx \|x - f(p) + J\delta_p\| = \|\epsilon - J\delta_p\|$. El valor de δ_p es obtenido por lo tanto cuando se cumple que $J^T J\delta_p = J^T \epsilon$.

La matriz $J^T J$ es la Hessiana aproximada. El método de LM resuelve una variación de esta Hessiana. Las ecuaciones se llaman normales aumentadas.

$$N\delta_p = J^T \epsilon$$

Aquí los elementos fuera de la diagonal son idénticos a los de $J^T J$ y los elementos de la diagonal vienen dados por $N_{ii} = \mu + [J^T J]_{ii}$ para $\mu > 0$. La estrategia para alterar los elementos de la diagonal se llama amortiguamiento y μ es llamado término de amortiguamiento. Si el parámetro actualizado $p + \delta_p$ calculado de $N\delta_p = J^T \epsilon$ reduce el error, la actualización es aceptada, procediendo a disminuir el termino de amortiguamiento.

El proceso de repetir las ecuaciones normales aumentadas $N\delta_p = J^T \epsilon$ para diferentes valores del término de amortiguamiento hasta que se obtiene una actualización del vector de parámetros se considera como una iteración del algoritmo LM.

Debido a que el término de amortiguamiento es ajustado cada iteración con el objeto de reducir el término de error ϵ, si el valor de este parámetro es alto, entonces la matriz N es básicamente diagonal y la actualización de δ_p está en la dirección de

pendiente más elevada (steppest descent direction). El parámetro de amortiguamiento sirve para manejar problemas numéricos cuando $J^T J$ es cercano a singular. El algoritmo LM es adaptativo ya que este controla el término de amortiguamiento. Por esta razón LM es capaz de alternar entre una aproximación suave cuanto está lejos del mínimo y una convergencia rápida cuando está en las proximidades del mismo.

El algoritmo LM termina cuando por lo menos una de las condiciones siguientes es cumplida:

- La magnitud del gradiente $J^T \epsilon$ cumple con la tolerancia.
- El cambio relativo de δ_p cumple con la tolerancia $\in_2$
- El error $\epsilon^T \epsilon$ cumple con $\in_3$
- Un numero de iteraciones máximas es alcanzado

Si una matriz de covarianza $\sum_x$ para el vector x es disponible, puede ser incorporada al algoritmo LM, minimizando $\epsilon^T \sum_x^{-1} \epsilon$. Así el problema será resuelto minimizando las ecuaciones normales ponderadas:

$$J^T \sum_x^{-1} J \ \delta_p = J^T \sum_x^{-1} \epsilon$$

2.7.2 Algoritmo Backpropagation

El algoritmo Backprogation [12] puede ser utilizado para entrenar cualquier tipo de red neuronal, sin embargo su aplicación más usual es en las redes llamadas Multilayer Perceptron. Estudiaremos a continuación el algoritmo Backpropagation aplicado a MLP.

Sea una red neuronal multicapa de L capas, en la cual el valor real de una neurona j de la capa l denominamos como x_j^l . Según esta convención las neuronas de la capa de ingreso son x_j^0 y las de la última capa son x_j^L. Los pesos sinápticos de las conectividades que salen de la capa l desde la neurona j hasta la neurona i de la siguiente capa son W_{ij}^l y el término umbral de esta misma capa l y neurona i es b_i^l .

Establecida la nomenclatura procedemos a describir el algoritmo Backpropagation.

1- Mapeo de x^0 a x^L . El vector de entrada x^0 es transformado en el vector x^L evaluando la siguiente ecuación:

$$x_i^l = f(u_i^l) = f\left(\sum_{j=1}^{nl-1} W_{ij}^l x_j^{l-1} + b_i^l\right) \qquad l = 1,2,\ldots\ldots,L$$

2- Cálculo del error del mapeo. La diferencia entre el valor objetivo d y el valor calculado x_j^L es calculado mediante:

$$\delta_i^L = f'(u_i^L)\,(d_i \; - \; x_j^L)$$

3- Error hacia atrás. El error en la salida de la red respecto al valor objetivo es repartido hacia atrás dependiendo de la función de activación y de los pesos sinápticos:

$$\delta_j^{l-1} = f'(u_j^{l-1}) \sum_{i=1}^{nl} W_{ij}^l \delta_i^l \qquad l = 1,2,\ldots\ldots,L$$

4- Actualización de pesos. Los pesos sinápticos y los términos umbral son actualizados utilizando los valores de x_j^l y δ_j^{l-1}mediante las ecuaciones siguientes:

$$\Delta W_{ij}^l = \eta \delta_i^l \; x_j^{l-1}$$

$$\Delta b_i^l = \eta \delta_i^l$$

Este proceso es una iteración del algoritmo. Una vez que se han actualizado los pesos y los términos umbral se procede a ingresar un nuevo vector x^0 para generar otra predicción y su consecuente actualización de pesos y términos umbral. El proceso sigue hasta que el valor del error de cálculo de la red cumpla con una tolerancia o llegue hasta un número límite de iteraciones.

Cuando los datos del vector x^0 son escogidos aleatoriamente del conjunto de datos escogidos para entrenamiento, los pesos resultantes al final del ingreso de estos datos son llamados pesos sinápticos de la época k. Si generamos un nuevo entrenamiento escogiendo nuevamente en forma aleatoria los datos del conjunto de entrenamiento, los pesos resultantes serán los de la época k+1. Así podemos generar varias épocas y comparar el error mínimo de cada época, para finalmente escoger los pesos de la época que ha generado el menor error respecto a otras épocas.

2.7.3 Métodos basados en gradientes conjugados

Estos métodos de segundo orden utilizan las segundas derivadas de la función objetivo. En contraste técnicas de aprendizaje como backpropagation utilizan derivadas de primer orden, por lo cual son llamadas técnicas de primer orden. Desde el punto de vista matemático las técnicas de segundo orden encuentran mínimos locales de una mejor manera que las de primer orden, sin embargo su costo computacional es más alto. El algoritmo de backpropagation busca el mínimo según la dirección del gradiente de la función objetivo, mientras que los métodos de gradientes conjugados lo busca en una dirección conjugada a las direcciones de los pasos anteriores.

Sea $\phi(x)$ una función multidimensional a ser minimizada.

Su desarrollo en serie de Taylor hasta el término cuadrático es:

$$\phi(x) \approx \phi(u) + D\phi^T[x-u] + \frac{1}{2}[x-u]^T D^2\phi[x-u]$$

Simplificando para u=0, y haciendo cambio nomenclatura, la condición de mínimo requiere que:

$$\nabla\phi(x) = D\phi \quad + D^2\phi[x] = 0$$
$$Ax - b = 0$$

Donde:
$D^2\phi = A\ es\ la\ matriz\ Hessiana$
$\nabla\phi(x) = -b\ es\ el\ valor\ negativo\ del\ gradiente$

Por lo tanto el problema de minimización de la función $\phi(x)$ se reduce al de la solución del sistema $Ax = b$, para el cual es necesario calcular la matriz Hessiana y el gradiente de la función a ser minimizada. Además, el vector x resultado de la solución del sistema es un punto crítico de la función $\phi(x)$.

El método de los gradientes conjugados (CG) es un método para resolver iterativamente sistemas de ecuaciones de la forma:

$$Ax = b$$

Donde x es el vector de incógnitas, A es una matriz de términos conocidos, y b es el vector independiente.
Las condiciones sobre la matriz A es que debe ser cuadrada, simétrica, y definida positiva. Una matriz es definida positiva si se cumple que:

$$x^T A\, x > 0$$

Ahora, supongamos que x^* es la solución real del sistema $Ax = b$, si empezamos con x_0 a buscar la solución, necesitamos saber cuan distantes estamos de la solución real. Para la solución del sistema se cumple que $Ax - b = 0$, por lo tanto $r_k = b - Ax_k$ el residuo será la métrica buscada que mide la diferencia entre la solución propuesta y la real.

Por otro lado, la búsqueda del punto mínimo se la realizará en direcciones ortogonales, debido a esto se llama gradientes conjugados. El proceso para encontrar direcciones ortogonales que podemos utilizar es el de ortonormalización de Gram-Schmith.

Este proceso es sucesivo y sigue la siguiente expresión:

$$p_{k+1} = r_k - \sum_{i \le k} \frac{p_i^T A\, r_k}{p_i^T A\, r_i}\, p_i$$

Si seguimos esta dirección, la siguiente ubicación óptima es:

$$x_{k+1} = x_k - \alpha_{k+1}\, p_{k+1}$$

$$\alpha_{k+1} = \frac{p_{k+1}^T\, r_k}{p_{k+1}^T A\, r_{k+1}}$$

Finalmente el algoritmo resultante es:

$r_0 = b - Ax_0$
$p_0 = r_0$
$k = 0$
Inicio de iteraciones:

$\alpha_{k+1} = \frac{p_{k+1}^T\, r_k}{p_{k+1}^T A\, r_{k+1}}$
$x_{k+1} = x_k - \alpha_k\, p_k$
$r_{k+1} = r_k - \alpha_k A\, p_k$
Si r_{k+1} es menor que una tolerancia sale de iteraciones
$\beta_k = \frac{r_{k+1}^T\, r_{k+1}}{r_k^T\, r_k}$
$p_{k+1} = r_{k+1} + \beta_k\, p_k$

k=k+1

Fin de iteraciones

CAPITULO 3

DEFINICIÓN DE LA ROBUSTEZ DE UN MODELO

CAPÍTULO 3

DEFINICIÓN DE LA ROBUSTEZ DE UN MODELO

3.1 Definición de Robustez del modelo

Esta tesis analiza la robustez del método de las redes neuronales para la predicción de caudales aplicado a la cuenca alta del Rio Paute. El análisis de la robustez se ha basado en un análisis de sensibilidad primero y posteriormente un análisis de incertidumbre.

Las variaciones de los datos de entrada pueden deberse a la variabilidad propia de las variables de entrada (suponiendo que no existe errores en los mismos), por un lado, pero además pueden ser afectadas por errores en lectura, transcripción, cálculo, etc.

Por esta razón se realiza un análisis de sensibilidad para identificar las variables a las cuales es más sensible el modelo, específicamente las estaciones a las cuales es más sensible el modelo. Luego, para el análisis de incertidumbre se calcula la incertidumbre del modelo como la suma de incertidumbres de dos fuentes. Una, la incertidumbre en predicción del modelo, y la segunda la propagación de incertidumbre por errores en los datos. Es importante recordar que si bien es cierto durante las dos últimas décadas se ha investigado sobre estos dos tópicos, al análisis de sensibilidad y el análisis de incertidumbre, no existe una definición científica única de ellos, y su procedimiento depende del campo de aplicación [17].

3.2 Análisis de sensibilidad

El modelo propuesto está orientado a respuesta en tiempo real, por lo tanto debemos efectuar el análisis de sensibilidad a la disponibilidad en tiempo real de las variables de entrada.

Debido a las condiciones actuales de operación de las estaciones de medición, es usual la falta de información de las diversas estaciones. Esta falta de información no es

deseable debido a que el modelo es distribuido, en consecuencia es necesario disponer de la información de las nueve estaciones de precipitación y una de caudal en forma coincidente. Por lo tanto es importante que estudiemos como las variables de entrada afectan a la variable de salida (predicción de caudal). El sistema debería ser robusto frente a faltantes de información, sin embargo pretendemos identificar un grupo de variables básicas para el funcionamiento adecuado del modelo.

Resumimos las variables de entrada en la tabla 2:

Número de estación	Denominación de Estación	Variable
1	Cancan_Soldados	precipitación
2	Chanlud	precipitación
3	El_Portete	precipitación
4	El Labrado	precipitación
5	Machangara_DJ_Chulco	precipitación
6	Matadero_en_Sayausi	precipitación
7	Tarqui_DJ_Cumbe	precipitación
8	Ucubamba_en_ETAPA	precipitación
9	Yanuncay_en_Pucan	precipitación
10	Tomebamba_en Ucubamba	caudal

Tabla 2 *Estaciones de precipitación y caudal de la cuenca alta del Paute*

3.2.1 Cálculo de la sensibilidad

Existen diferentes métodos para calcular la sensibilidad de un modelo. Además dependiendo de las características del modelo tratado esta sensibilidad puede ser determinada respecto a los parámetros y respecto a las variables de ingreso [15]. En el modelo con el cual se trabaja en esta tesis las variables de entrada serán objeto de estudio.

Primero se calculará el MSE en predicción del modelo con todas las estaciones trabajando simultáneamente. Obviamente se espera que el MSE resultante sea el menor esperado. Luego se procederá a simular la falta de datos de cada una de las estaciones en forma secuencial para así determinar los respectivos MSE_i en predicción.

El cálculo de la sensibilidad del modelo a cada estación es:

$$S_i = \frac{MSE_i}{MSE}$$

Donde:

S_i: Sensibilidad del modelo a la estación i
MSE_i: MSE del modelo en predicción sin la estación i
MSE : MSE del modelo en predicción con todas las estaciones.

De la formula anterior podemos inferir que el S_i necesariamente es mayor o igual que la unidad, debido a que el menor valor de MSE es el del modelo funcionando con todas las estaciones. Ordenando las estaciones en función de la sensibilidad de la mayor a la menor podemos identificar las estaciones de mayor importancia para el modelo.

3.3 Análisis de incertidumbre

Un análisis de incertidumbre generalmente es complementado con un análisis de sensibilidad. El análisis de sensibilidad determina como la variación de las variables o parámetros del modelo afectan a la respuesta del modelo, y de esta manera identificar las más importantes. Contrariamente el análisis de incertidumbre determina la forma como se propaga la incertidumbre ya sea de los datos, del modelo o submodelos a través del modelo, y se refleja en incertidumbre de la respuesta del modelo, en este caso la predicción del caudal [17].

El análisis de sensibilidad específicamente nos ayudará a determinar la robustez de la predicción de los caudales frente a hipótesis de funcionamiento, como podrían ser el funcionamiento de la red con todas las estaciones o por el contrario funcionando con las estaciones más influyentes.

Aunque existe un consenso respecto de la determinación probabilística de la incertidumbre de un fenómeno, no existe un método establecido para obtener sus distribuciones, y demás parámetros estadísticos. Tampoco se ha establecido específicamente una metodología para identificar las fuentes de incertidumbre.

Varias autores [17] proponen las siguientes consideraciones en cuanto a la identificación de fuentes de incertidumbre, aplicados a la estimación hidrológica y la modelación hidráulica:

- Contexto del modelo: identificación correcta de las variables y parámetros que influyen en el fenómeno.

- Incertidumbres en la implementación del modelo: en el proceso de implementar el modelo desde la percepción del fenómeno a su conceptualización se pueden incluir incertidumbres de manera involuntaria.

- Propagación de incertidumbre en el modelo: la influencia en la respuesta del modelo frente a incertidumbres en los datos de entrada es importante de considerar.

- Incertidumbre en los parámetros del modelo.

- Incertidumbre dependiente de la metodología de estimación: debido a que existen diversos métodos de estimación de incertidumbre, su cuantificación no será única. La selección del método dependerá de la disponibilidad de datos, el tipo de variables, la escala de las variables, la distribución del error de la estimación de las variables, etc.

- Incertidumbre en las mediciones.

- Incertidumbre en la estabilidad de condiciones futuras: un ejemplo actual son los diversos escenarios que presentan los modelos de circulación global.

3.3.1 Un método probabilístico para generación de la incertidumbre

Para el análisis de incertidumbre se ha elegido un método probabilístico. La fuente de incertidumbre estudiada es en los datos de entrada, o variables. Este modelo no posee parámetros, es por esto que no se ha incluido incertidumbre en los parámetros.

La estimación de la incertidumbre o error en los datos de entrada en un aspecto importante a ser determinado. Estudios previos sostienen que la desviación estándar del error de medición de precipitación está alrededor de 0.8 mm [2]. Por esto se ha generado una distribución normal $N(\mu = \bar{p}; \sigma = 0.8)$ con precipitación media y desviación estándar 0.8 mm, para luego propagarla por el modelo.

Ya que este modelo es distribuido, y debido a que el análisis de sensibilidad se ha realizado respecto a las estaciones, el análisis de incertidumbre se lo ha realizado respecto a las estaciones también. Queda como tarea posterior el estudio de estos dos análisis respecto al día de influencia.

Debido a que las respuestas de este modelo no pueden calcularse analíticamente, la generación de la distribución normal de incertidumbre en las variables de entrada será aleatoria, y se estudiará la respuesta del modelo. Podemos llamar a este método probabilístico de Monte Carlo.

La elección de la normalidad para la distribución de la incertidumbre está respaldada por un análisis del error de predicción. Varias pruebas de normalidad aplicadas a los errores de predicción del modelo categorizan esta distribución como normal (Fig. 14).

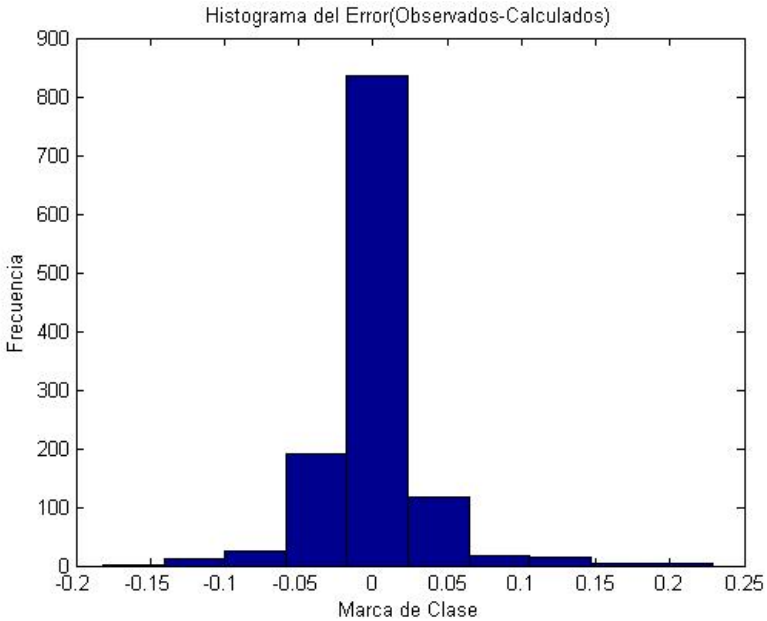

Figura 14 *Histograma de los errores de predicción del modelo*

3.3.2 Propagación de la incertidumbre por la descomposición de la varianza

Debido a la normalidad de la distribución de los errores de predicción del modelo se ha escogido el método de la descomposición de varianzas para dividir la incertidumbre total en sus diferentes fuentes [17]. Este método considera que las fuentes de incertidumbre son independientes, y se pueden expresar mediante:

$$O + E_o = Y(P, E_P, X, E_X) + E_\gamma\ (P, E_P, X, E_X) + E_R$$

O : Vector de observaciones
Y : Vector de valores modelados
P : Vector de parámetros
E_o : Error de observación
E_P : Error de los parámetros
E_X : Error de las entradas o variables
E_γ : Error del modelo
E_R : Error aleatorio

El residuo entre los valores observados y calculados por el modelo es:

$$Y(P, E_P, X, E_X) - O = E_o + E_\gamma\ (P, E_P, X, E_X) + E_R$$

Los signos de los errores no interesan pues pueden ser en cualquier sentido.

Para una generación aleatoria normal de la incertidumbre se cumple [17]:

$$\sigma_Y^2 = \sum_{i=1}^{nx} \sigma_{YXi}^2 + \sum_{i=1}^{np} \sigma_{YPi}^2 + \sum_{i=1}^{nF} \sigma_{YFi}^2$$

σ_Y^2 : Varianza que refleja la incertidumbre de la variable Y
σ_{YXi}^2 : Varianza de genera la incertidumbre de la variable Xi.
σ_{YPi}^2 : Varianza que genera la incertidumbre del parámetro Pi.
σ_{YFi}^2 : Varianza que genera la incertidumbre del submodelo Fi.

En el modelo estudiado, la incertidumbre de los parámetros puede aplicarse a la incertidumbre de los pesos sinápticos de la red. Debido a la robustez del método ante

la variación de la arquitectura y debido a la falta de conocimiento en la interpretación física de los pesos, un estudio específico de esta incertidumbre no se ha realizado. Otro argumento para no considerar la incertidumbre de los pesos sinápticos es la amplia distribución espacial de estos en las diferentes arquitecturas, sin embargo las respuestas de los modelos son muy similares, significa que la variación de los pesos se compensan y por lo tanto no son fenómenos independientes estadísticamente, como es la incertidumbre de los datos de medición en las diferentes estaciones. La incertidumbre de los submodelos se ha calculado como la incertidumbre del modelo mismo en predicción. Por lo tanto se ha considerado solamente incertidumbre calculada como la varianza generada por la variable Xi más la varianza en predicción del modelo. Basados en estos argumentos, limitamos nuestro estudio de la incertidumbre a dos términos σ_{YFi}^2 que es la incertidumbre de submodelos que será redefinida como σ_M^2 y a $\sum_{i=1}^{9} \sigma_{YXi}^2$ que es la incertidumbre inducida en el modelo por la incertidumbre de medición de precipitación en las 9 estaciones.

$$\sigma_M^2 = \sigma_{YFi}^2$$

$$\sigma_Y^2 = \sum_{i=1}^{9} \sigma_{YXi}^2 + \sigma_M^2$$

Las estaciones a las que se ha aplicado incertidumbre son las estaciones de lluvia. No se ha estudiado la propagación de incertidumbre de la estación de caudal por la relación de la predicción del caudal con valores de días anteriores del caudal.

PARTE 2

IMPLEMENTACIÓN DE UN MODELO DE REDES NEURONALES Y ANÁLISIS DE ROBUSTEZ

CAPITULO 4

PRE PROCESAMIENTO DE LOS DATOS

CAPITULO 4

PRE PROCESAMIENTO DE LOS DATOS

4.1 Definición de los datos

Debido a que el modelo desarrollado es distribuido es necesario que la información sea coincidente en el tiempo. Es decir, las series de tiempo de las variables de entrada deben ser completas en las fechas o intervalos de tiempo escogidos. Esto ha dificultado la disponibilidad de series de tiempo prolongadas. En el caso de modelos concentrados es posible hacer una ponderación de los datos de las estaciones faltantes mediante técnicas estadísticas. Para el modelo que se ha estudiado en esta tesis es necesario información de cada estación, debido a que el modelo es altamente sensible a la información de cada estación, desde luego esto es una debilidad y una fortaleza a la vez.

Las fechas de datos coincidentes se presentan a continuación:

INTERVALOS DE REGISTRO DE PRECIPITACIÓN Y CAUDAL		
Intervalo 1	Desde: Enero 2 del 2002	Hasta: Agosto 1 del 2003
Intervalo 2	Desde: Marzo 20 del 2008	Hasta: Diciembre 30 del 2009

Tabla 3 *Fechas de datos coincidentes de información*

Lo visto en el párrafo anterior es una descripción temporal de los datos utilizados. La cantidad de información disponible es de 1228 días. En cuanto a la distribución espacial es importante recalcar que la calidad del modelo depende mucho de la ubicación adecuada de las estaciones de registro, especialmente de las estaciones de registro de precipitación ya que los puntos escogidos deben ser representativos dentro de una subcuenca. Muchas veces estos puntos no pueden ser ubicados en los lugares óptimos por asuntos logísticos, de seguridad o mantenimiento. Se ha partido de la hipótesis que los datos disponibles no han sido afectados por equivocaciones y son representativos de cada zona.

La ubicación de las estaciones se presenta en la Fig. 15. Las estaciones de registro de precipitación están señaladas con triángulos, las de caudal con círculos. La salida de la

parte alta de la cuenca está encerrada con un círculo que representa la estación de medición de caudal de Ucubamba en el Tomebamba.

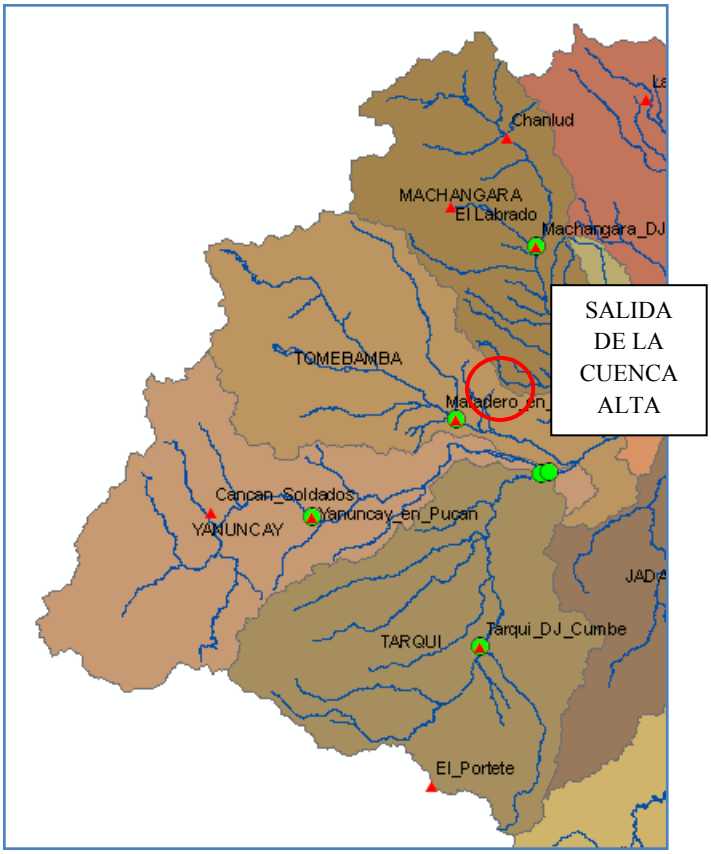

Figura 15 *Distribución espacial de las estaciones de precipitación y caudal*

Las estaciones tanto de precipitación como de caudal que se han utilizado en esta tesis están en la tabla 4. Las lluvias están expresadas en milímetros y los caudales en m3/s.

Tabla 4 *Estaciones de precipitación y caudal, unidades, pesos, ubicación*

Nombre	Cancan_Soldados	Chanlud	El_Portete	El Labrado	Machangara_DJ_Chulco	Matadero_en_Sayausi	Tarqui_DJ_Cumbe	Ucubamba_en_ETAPA	Yanuncay_en_Pucan	Tomebamba_en Ucubamb
Id Estación	0000010	0000020	0000030	0000070	0000080	0000090	0000110	0000120	0000130	0000008000
Captor	Lluvia	Lluvia	Lluvia	Lluvia	Lluvia	Lluvia	Lluvia	Lluvia	Lluvia	Caudal
Unidad	(mm)	(mm)	(mm)	(mm)	(mm)	(mm)	(mm)	(mm)	(mm)	(m3/s)
Latitud	-2.94	-2.68	-3.13	-2.73	-2.75	-2.88	-3.04	-2.87	-2.94	-2.87
Longitud	-79.24	-79.03	-79.08	-79.07	-79.01	-79.07	-79.05	-78.94	-79.17	-78.94
Fecha	1	2	3	4	5	6	7	8	9	10
02/01/2002 12:00	9.00	2.00	4.50	5.50	23.50	5.10	0.60	0.90	0.80	51.62
03/01/2002 12:00	9.60	1.10	1.20	2.60	5.60	0.50	0.10	0.50	0.30	37.23
04/01/2002 12:00	0.90	2.70	1.60	4.50	5.50	0.40	0.10	1.10	0.20	25.73
05/01/2002 12:00	0.20	0.50	2.80	2.00	7.00	0.20	0.00	2.00	0.10	21.63
06/01/2002 12:00	0.20	4.80	0.40	4.70	1.50	0.20	0.10	0.70	2.70	17.38
07/01/2002 12:00	0.60	11.00	8.00	0.90	2.00	0.10	0.10	0.20	0.40	14.97
08/01/2002 12:00	0.10	0.20	1.10	2.00	1.90	0.10	0.00	0.10	0.10	13.51
09/01/2002 12:00	0.10	2.20	0.50	6.80	0.20	0.20	0.10	0.10	0.20	11.74
10/01/2002 12:00	0.10	0.20	0.20	0.20	0.10	0.10	0.20	0.20	0.10	10.23
11/01/2002 12:00	0.80	0.20	0.10	0.10	0.30	1.20	0.10	4.70	1.30	8.88
12/01/2002 12:00	5.10	3.20	0.10	2.30	0.50	2.50	0.10	1.60	2.90	8.54
13/01/2002 12:00	0.50	4.90	5.90	2.40	8.90	5.90	0.20	4.30	0.20	9.04
14/01/2002 12:00	0.30	2.80	0.80	5.30	0.50	1.00	6.90	0.20	1.30	8.93
15/01/2002 12:00	0.20	0.40	0.40	0.40	0.00	0.30	0.20	0.20	0.20	9.10
16/01/2002 12:00	8.80	6.80	1.40	0.30	0.10	21.70	5.80	0.10	7.10	7.98
17/01/2002 12:00	3.20	0.50	1.80	0.00	0.10	1.10	0.00	0.00	2.60	**8.98**
18/01/2002 12:00	0.40	0.10	0.50	0.10	0.00	0.40	0.10	0.00	0.10	7.10

4.2 Escala de datos

De la tabla anterior podemos ver que las escalas de los datos de precipitación y de caudal difieren considerablemente. Como veremos más adelante estos datos serán las variables de entrada para la red neuronal, por lo tanto es aconsejable que sus escalas sean comparables ya que las funciones de activación utilizadas son del tipo sigmoidal, las cuales requieren que estén dentro de un rango cercano a [0; 1] que es el rango de la función sigmoidal [18].

Para cambiar la escala de los datos de precipitación dentro del rango [-3;3] se ha utilizado la ecuación:

$$p^* = \frac{p * 6}{130} - 3$$

Siendo:
p^*: Precipitación re escalada
p : Precipitación registrada

Para cambiar la escala de los datos de caudal dentro del rango [0.1; 0.9] se ha utilizado la ecuación:

$$q^* = \frac{q * 0.8}{200} + 0.1$$

Siendo:
q^*: Caudal re escalado
q : Caudal registrado

Luego del cálculo de la red neuronal se han revertido los datos a su escala original, mediante las ecuaciones:

$$\mathrm{p} = \frac{130 * (\mathrm{p}^* + 3)}{6}$$

$$\mathrm{q} = \frac{200 * (\mathrm{q}^* - 0.1)}{0.8}$$

Otra razón importante para escalar los datos adecuadamente es la rapidez de aprendizaje de la red, e incluso evitar inconvenientes en el entrenamiento de la red.

4.3 Identificación de variables de entrada

Si bien es cierto que las variables que están involucradas en el problema de predicción de caudales son precipitación, temperatura, evapotranspiración, evaporación y otras más, los datos disponibles de la parte alta de la cuenca son los de precipitación en diversos puntos de la cuenca, y registro de caudal en el punto de Ucubamba en el rio Tomebamba.

Debido a que la precipitación no se refleja instantáneamente como caudal en el río, por el contrario esta precipitación es retenida en los diferentes componentes de caudal dentro de una cuenca, es importante identificar el número de días de historia que están afectando a la medición de caudal presente.

Para identificar el número de días de precipitación que se correlacionan con el caudal que se desea predecir aplicamos la función de correlación cruzada. La correlación cruzada estima la correlación de una variable respecto a otra considerando también retrasos. Observando la Fig. 16 podemos identificar algunas estaciones que tienen mayor correlación que otras. Según un umbral de 0.5 de correlación la mayoría de estaciones mantienen una correlación mayor durante los 10 primeros días. Por esta razón se ha decidido utilizar 10 días de historia para precipitación. Esta elección es razonable debido a que por otro lado una elección de más días de historia demandaría demasiado costo computacional.

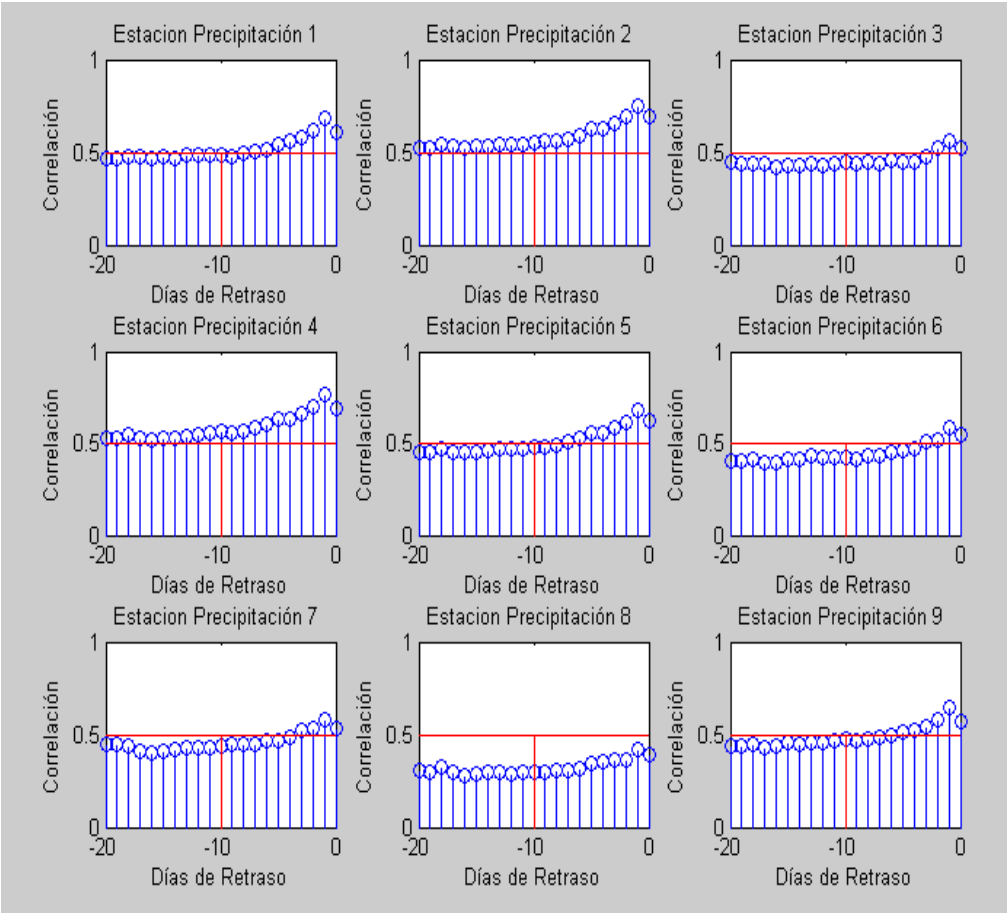

Figura 16 *Grafico correlación cruzada de estaciones de precipitación con la de caudal*

Para determinar los días de retraso del caudal se ha utilizado la función de auto-correlación, como se puede observar en la Fig. 17. La auto-correlación es una correlación cruzada de una serie con sí misma. Esta función se utiliza para identificar patrones tales como la presencia de señales periódicas afectadas por ruido, identificar una frecuencia fundamental, etc.

Debido a que según los valores de auto-correlación del caudal se mantienen por encima del umbral que está representada por la lineal azul en la Fig. 17, hemos elegido una historia de 10 días como un periodo razonable.

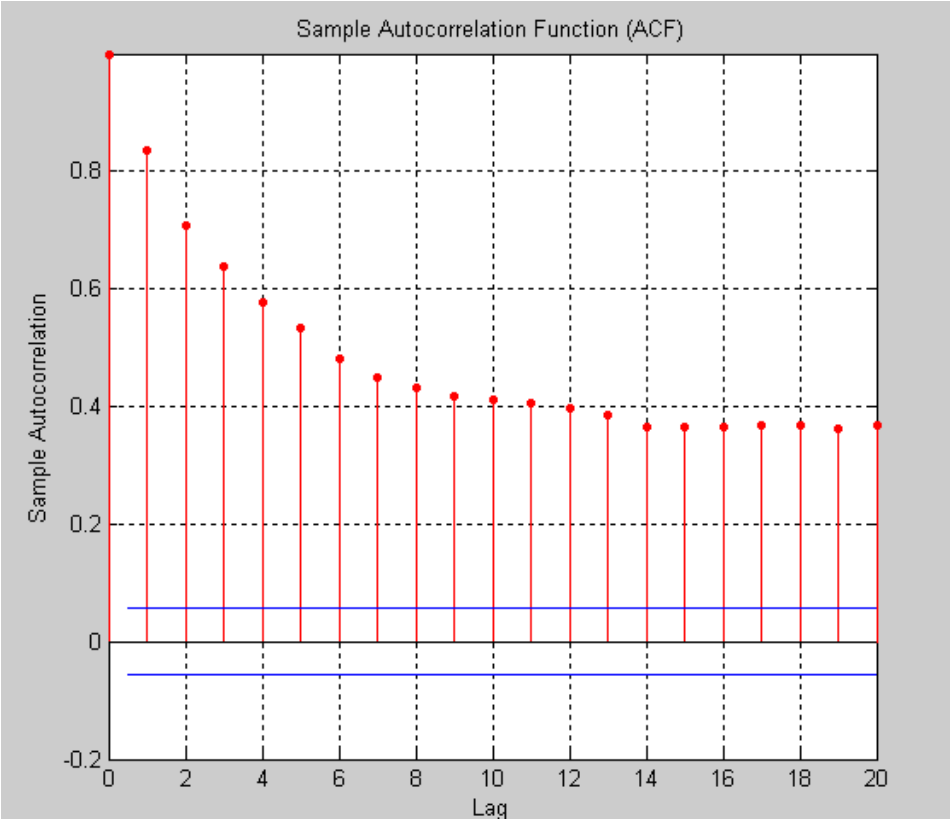

Figura 17 *Grafico de la auto-correlación del caudal*

Como se mostró en la sección 4.1 existen 9 estaciones de precipitación y una de caudal. La estación de caudal sirve también como variable de entrada para predicción de caudal con un día de retraso. En la tabla 4 podemos ver los datos de las diez estaciones desde la fecha 07/01/2002 hasta 16/01/2002 como datos que influencian el caudal medido la fecha 17/01/2002. Es decir, los 10 días pasados de las 10 estaciones constituyen 100 datos de ingreso para calcular el caudal del día siguiente.

Por el gran número variables de las cuales depende el modelo se analizó la información mediante el método de Análisis de Componentes Principales (PCA). Este método pretende encontrar las direcciones principales de un conjunto de varias variables, y de esta manera identificar las combinaciones lineales que generan las componentes más importantes del fenómeno. El problema que nos ocupa se plantea de

la siguiente manera. Dadas 10 estaciones de registro, hallar la combinación lineal de ellas tal que la información de registro se maximice utilizando esas direcciones principales.

Las direcciones principales se identifican mediante el orden del valor propio de la matriz de covarianza. El valor propio mas grande representa a la dirección 1, el siguiente a la dirección 2, así sucesivamente (Fig. 18). En la tabla 5 se presenta los resultados de los valores propios, así como el porcentaje de varianza que ellos representan. En la tabla 5 podemos ver la primera componente represente el 51.4% de la varianza, hasta la sexta componente se acumula un total del 88.4% de la varianza, lo cual es aceptable considerando que podríamos eliminar errores de las medidas y ruido tomando solo 6 componentes. No se ha utilizado el método de las componentes principales debido a que el software tiene que operar en tiempo real, y cada vez que cambia la tendencia de precipitación se debería hacer un PCA para evaluar los coeficientes de la combinación lineal que intervienen en las estaciones

Valores propios:

	F1	F2	F3	F4	F5	**F6**	F7	F8	F9	F10
Valor propio	5.14	1.01	0.76	0.73	0.66	**0.53**	0.40	0.35	0.25	0.16
% varianza	51.40	10.10	7.62	7.33	6.63	**5.28**	3.97	3.51	2.52	1.63
% acumulado	51.40	61.49	69.12	76.45	83.08	**88.36**	92.33	95.84	98.37	100.00

Tabla 5 *Valores propios de la matriz de covarianza.*

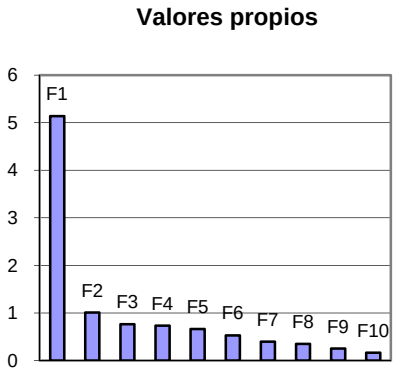

Figura 18 *Grafico de valores propios*

4.4 Análisis de resultados

En la sección 4.1 Definición de los datos, se estudió la disponibilidad de los datos y los requerimientos de datos que el modelo necesita, por ser de naturaleza distribuido.

Las series de tiempo necesarias para el modelo deben ser coincidentes en temporalidad, con datos de precipitación y caudal diarios. Las series de tiempo más largas disponibles son de 1228 días, los cuales representan a registros no continuos desde el 2 de enero del 2002 hasta el 1 de agosto del 2003 para el primer intervalo y desde el 20 de marzo del 2008 hasta el 30 de diciembre del 2009.

En la sección 4.2 Escala de datos, se ha estudiado la escala a la que los datos deben ser transformados. Debido a que el modelo trabaja con precipitaciones y caudales, sus escalas originales son diferentes, por lo cual se cambio a una escala de (-3; 3) para precipitaciones y (0.1; 0.9) para caudales. Utilizar la escala adecuada permitió el uso de funciones de activación tipo sigmoidales, mejoró las velocidades de entrenamiento, y aportó a la disminución de errores numéricos.

En la sección 4.3 Identificación de variables de entrada, se establece la historia de 10 días de las variables que afectan a la predicción con un día de retraso, mediante los criterios de correlación cruzada del caudal con las estaciones de precipitación y de auto-correlación del caudal. Debido a que existen 9 estaciones de precipitación y 1 de caudal, cada una de estas con una historia de 10 días, el modelo se alimenta de 100 variables de entrada.

Una estrategia para disminuir estas entradas directas al modelo es la de análisis de componentes principales, PCA. Del estudio realizado de PCA, se concluyó que mediante la combinación lineal de las 10 estaciones se puede calcular 6 estaciones principales las cuales alimentarían al modelo, y las cuales representan el 88.36% de la varianza de los datos. Sin embargo, no se ha procedido a la implementación del modelo mediante el uso de las componentes principales por condiciones operativas de funcionamiento en tiempo real, necesitando registros más prolongados para identificar plenamente las direcciones principales.

CAPITULO 5

IMPLEMENTACIÓN DE ARQUITECTURAS DE REDES NEURONALES

CAPÍTULO 5

IMPLEMENTACIÓN DE ARQUITECTURAS DE REDES NEURONALES

5.1 Toolbox de Matlab

En primera instancia se programó un Multilayer Perceptron (MLP) en Matlab, con dos capas ocultas, y con Back Propagation Error como algoritmo de aprendizaje. El aprendizaje era bastante lento y debido a que cualquier cambio en el programa tomaba mucho tiempo, se decidió que sería mejor probar arquitecturas en un programa comercial flexible y versátil, y luego cuando haya sido definida la arquitectura más adecuada, se podría proceder a la programación o compra de la licencia.

El programa elegido fue Matlab debido a que tiene un Toolbox de redes neuronales que es bastante general y flexible.

Este toolbox permite cambiar el número de capas, número de neuronas, su función de activación, diversos tipos de interconexión entre capas, diversas funciones de error para optimizar la red, varios tipos de algoritmos de aprendizaje. Todas estas opciones son programables.

La interfaz es bastante amigable, permite trabajar con ventanas, desde teclado o con scripts. El manejo de archivos tiene la flexibilidad característica del entorno Matlab.

Una opción que hemos utilizado del toolbox es la diferenciación interna entre muestra de entrenamiento, validación y predicción o test. La red se entrena con los datos de entrenamiento y va calculando el mse. Debido al aprendizaje en cada iteración, este mse de entrenamiento va disminuyendo. Paralelamente el programa va calculando el mse con los datos de validación que globalmente van disminuyendo hasta cierto punto y luego aumentan paulatinamente. En ese punto Matlab automáticamente detiene el entrenamiento de la red, debido a que la red se está sobreentrenando para los datos de entrenamiento, pero se está distanciando de datos que no hayan intervenido en su entrenamiento. Esta característica se llama “early stopping” y evita la especialización de la red, lo cual es poco deseable para la generalización deseada. Una vez que la red

ha detenido el entrenamiento, se procede a calcular el mse de los datos de predicción o test. Esta característica no poseen otros programas (por ejemplo Neural Machine).

5.2 Tipos de redes estudiadas

Para identificar la red óptima para el problema que ocupa esta tesis, se ha trabajado con el número de capas, numero de nodos o neuronas por capa, los tipos de conexión entre capas y neuronas entre capas y las funciones de transferencia utilizadas. La versatilidad que ofrece el toolbox de Matlab ha permitido probar varias arquitecturas de red.

1- **Red lineal**: la opción más sencilla es una red neuronal con una neurona. Sin embargo al utilizar una neurona, el problema idóneo a tratar es uno que posea una relación lineal entre las variables ò datos de entrada con los de salida. El problema precipitación escorrentía es altamente no lineal, sin embargo se ha obtenido resultados de la aplicación de una red lineal. (Ver Fig.19)

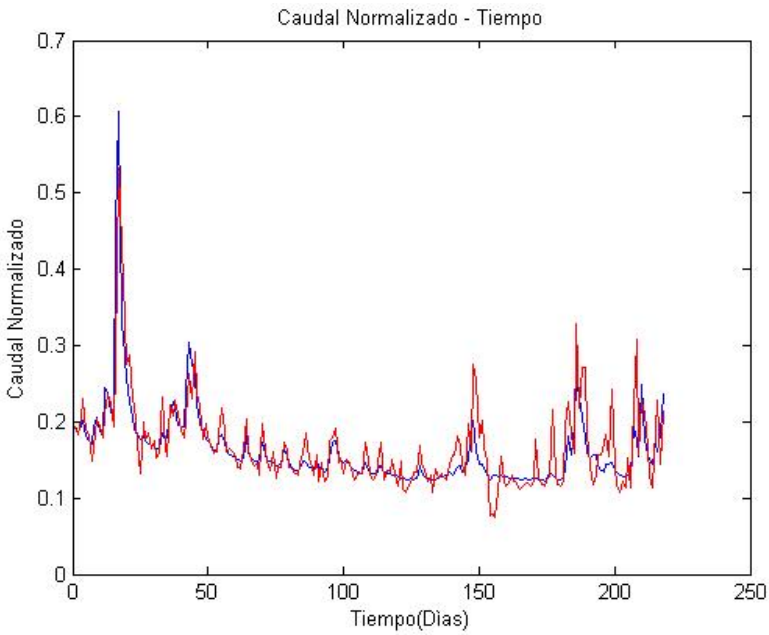

Figura 19 *Predicción red lineal*

2- **MLP:** Multilayer Perceptron, es una red de varias capas básicamente. Las capas son la de entrada, una o varias intermedias u ocultas y la de salida. La bibliografía recomienda por lo menos una capa oculta que no sea lineal, para posibilitar el tratamiento adecuado de problemas no lineales. Como referencia para nuestros prototipos en Matlab hemos utilizado el software Neural Machine, en el cual la red MLP utiliza una capa oculta y una de salida, además de la de entrada. Este programa tiene dos opciones respecto a funciones de activación, la función sigmoidal y la función lineal. Estas dos opciones se pueden aplicar a la capa oculta y a la capa de salida.
Mediante el toolbox de Matlab se puede optar por varias capas ocultas, pero de acuerdo a referencias bibliográficas, esto podría llevar a una red muy específica o sobreentrenada.

La primera red posee la capa de entrada, una capa oculta sigmoidal y la de salida lineal. La primera capa con 100 neuronas para un modelo con 10 estaciones y diez días de historia. En la capa oculta se ha experimentado con diversos números de neuronas, empezando desde 10 hasta 30. La capa de salida posee una neurona con función de transferencia lineal. (Figs. 20 y 21)

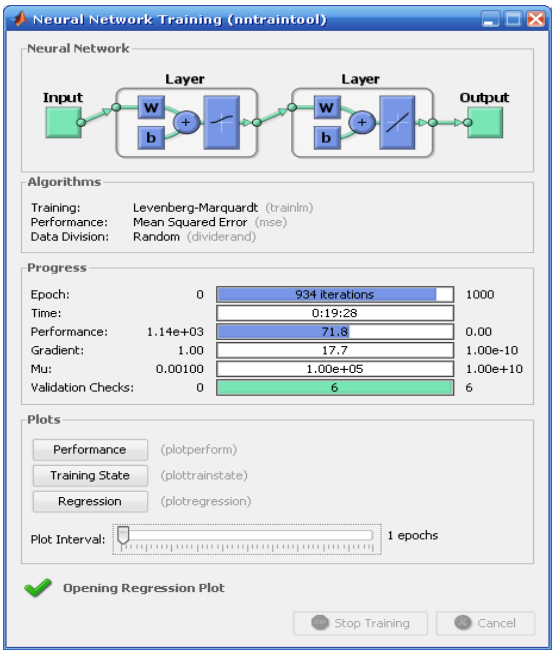

Figura 20 *Pantalla Interfaz durante entrenamiento de la red 2 capas*

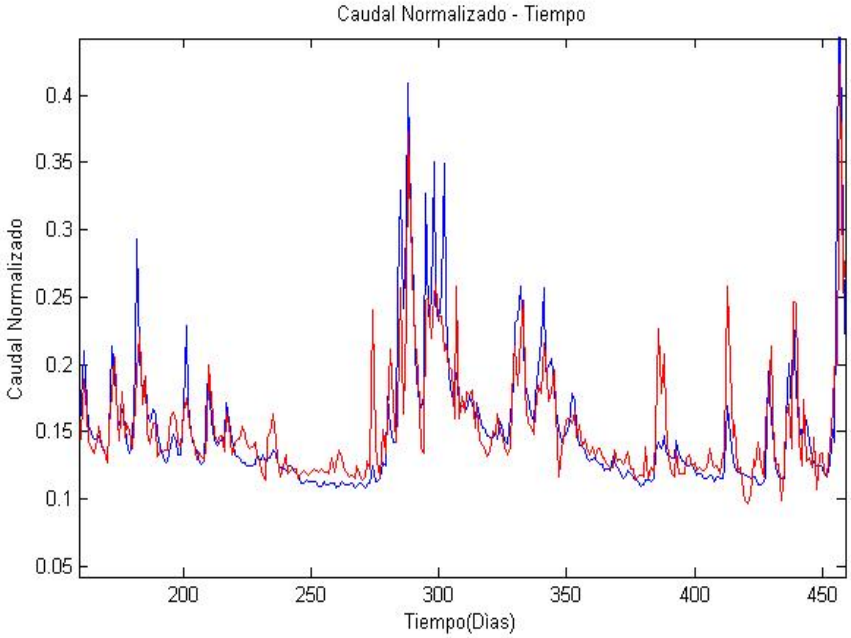

Figura 21 *Predicción MLP capa sigmoidal y capa lineal*

La segunda red MLP posee una capa de entrada, una capa oculta sigmoidal conectada a la de salida y a la siguiente oculta sigmoidal. Otra capa oculta sigmoidal conectada a la de salida y finalmente la de salida. La capa de entrada con 100 neuronas, la primera oculta sigmoidal con 10 neuronas, la segunda oculta sigmoidal con 5 neuronas, y finalmente la de salida con 1 neurona. (Figs. 22 y 23)

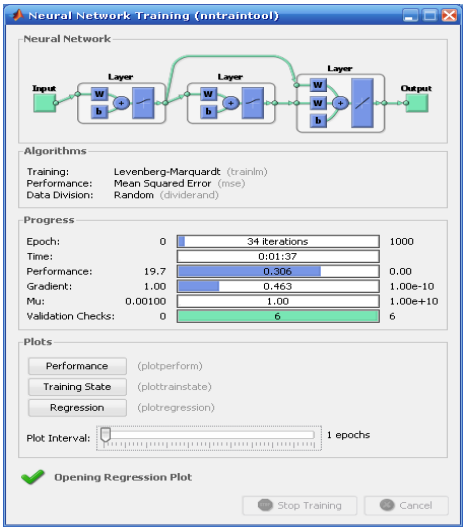

Figura 22 *Pantalla Interfaz durante entrenamiento de la red MLP 3 capas*

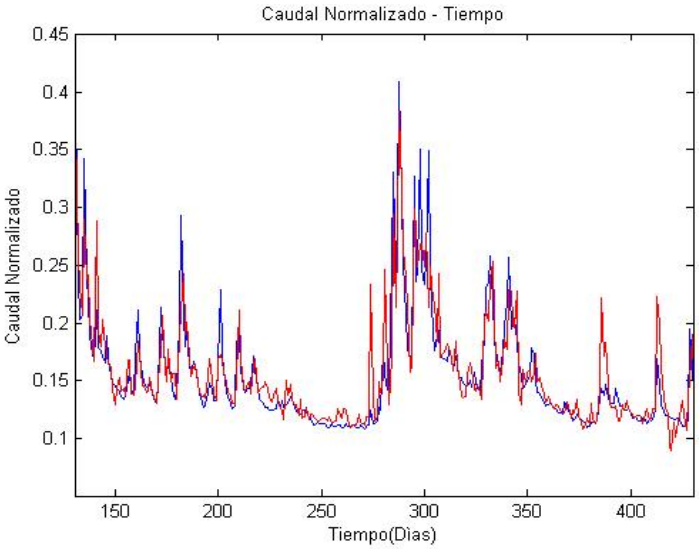

Figura 23 *Predicción MLP 2 capas sigmoidal y capa lineal*

La tercera red es la que presentamos a continuación. El número de neuronas en la primera capa oculta es 20, la siguiente es 10, la siguiente 5, y finalmente la de salida 1. Todas las capas son sigmoidales excepto la final que es lineal. (Figs. 24 y 25)

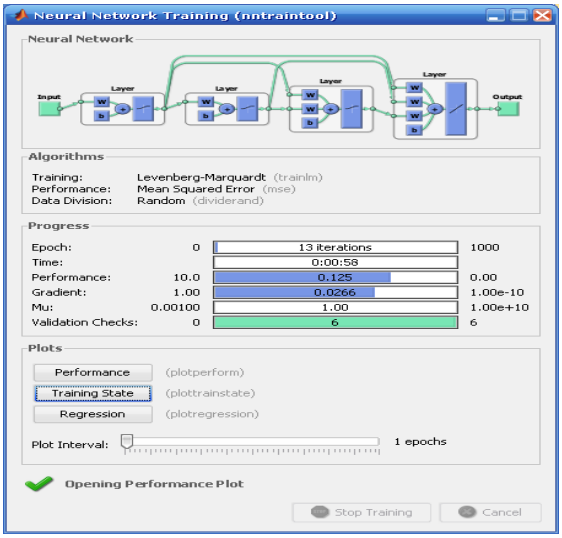

Figura 24 *Predicción MLP 3 capas sigmoidal y capa lineal*

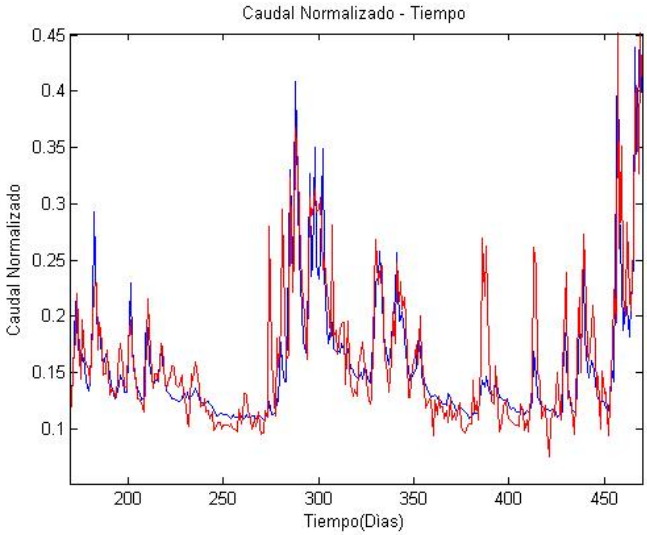

Figura 25 *Predicción MLP 3 capas sigmoidal y capa lineal*

5.3 Identificación de la red optima

Se ha realizado un programa que nos permita optimizar el parámetro Q^2 respecto al espacio de las capas y el número de neuronas.

La definición del parámetro estadístico Q^2 tiene el sentido del parámetro R^2 de Pearson pero aplicado para predicción. El error total respecto a la media es calculado en la etapa de entrenamiento, y el error del modelo es calculado en la etapa de predicción, la unidad menos la relación de errores definen el parámetro Q^2. (Fig. 26) [19]

$$Q^2 = 1 - \frac{PRESS}{TSS}$$

$$PRESS = \sum_{i=1}^{m}(q_{Pred}^{Obs} - q_{Pred}^{Cal})^2 \quad TSS = \sum_{i=1}^{n}(q_{Entren}^{Obs} - \bar{q}_{Entren}^{Cal})^2$$

Como se ha expuesto en secciones anteriores el número de datos disponibles es de 1228 días de información. Para la etapa de entrenamiento y validación se han utilizado 1000 días datos y para la etapa de predicción se han utilizado se han utilizado 228 días. En la figura a continuación se puede visualizar las dos etapas.

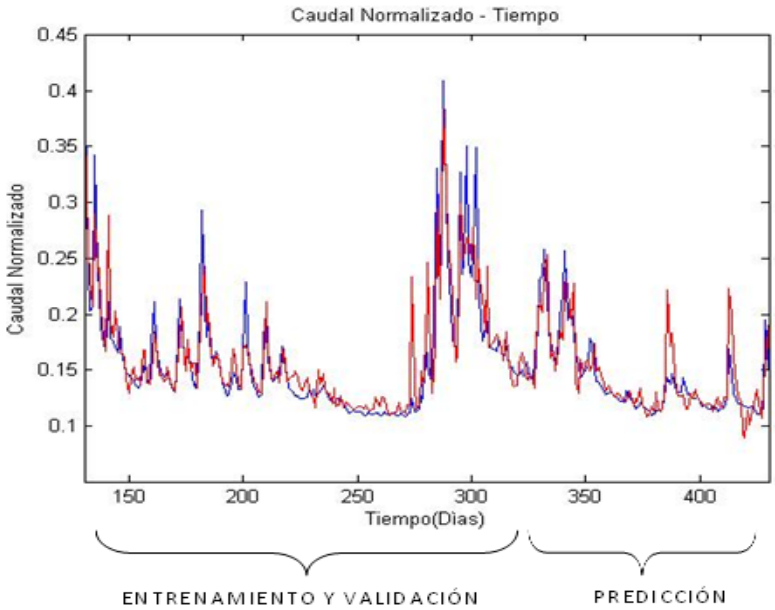

Figura 26 *Modelo en entrenamiento, validación y predicción*

5.3.1 Red una capa oculta sigmoidal y capa de salida sigmoidal

La primera arquitectura posee una capa oculta con función de activación sigmoidal y la capa de salida con función de activación sigmoidal. Se han corrido 8 tratamientos, es decir, 8 redes, una para cada tratamiento. Como podemos ver en la tabla 6 el parámetro que optimizamos en esta arquitectura es el número de neuronas en la capa oculta. Se ha experimentado con 3, 5, 10, 15, 20, 25, 30, 35. La capa de salida tiene 1 neurona pues se tiene un solo valor objetivo. Debido a que la etapa de entrenamiento y validación utiliza datos en forma aleatoria, esta respuesta Q^2 es promedio de tres corridas. Es decir que para generar los resultados de la tabla 6 se han corrido 3 veces cada tratamiento y luego el promedio de la respuesta Q^2 es el resultado expuesto. En forma gráfica representamos estos resultados en la Fig. 27.

Esta red presenta una respuesta promedio óptima para los tratamientos dos y tres, que corresponden a las siguientes configuraciones:

tratamiento	capa oculta 1	capa salida	respuesta
	sigmoidal	sigmoidal	Q^2
0	3	1	0.969
1	5	1	0.971
2	**10**	**1**	**0.979**
3	**15**	**1**	**0.978**
4	20	1	0.977
5	25	1	0.968
6	30	1	0.977
7	35	1	0.975

Tabla 6 *Valores de Q^2 red 5.3.1*

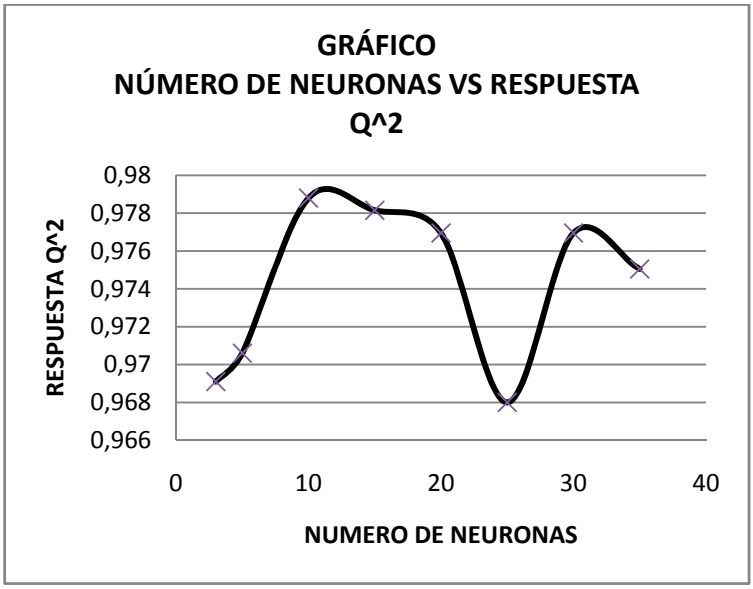

Figura 27 *Respuesta Q^2 vs número de neuronas red 5.3.1*

5.3.2 Red una capa oculta sigmoidal y capa de salida lineal

La segunda arquitectura posee una capa oculta con función de activación sigmoidal y la capa de salida con función de activación lineal. Se han corrido 9 tratamientos, es decir, 9 redes, una para cada tratamiento. Como podemos ver en la tabla 7 el parámetro que optimizamos en esta arquitectura es el número de neuronas en la capa oculta. Se ha experimentado con 3, 5, 10, 15, 20, 23,25, 27, 30. La capa de salida tiene 1 neurona pues se tiene un solo valor objetivo. Debido a que la etapa de entrenamiento y validación utiliza datos en forma aleatoria, esta respuesta Q^2 es promedio de tres corridas. Es decir que para generas los resultados de la tabla 5.3.2 se han corrido 3 veces cada tratamiento y luego el promedio de la respuesta Q^2 es el resultado expuesto.

Esta red presenta una respuesta promedio óptima para el tratamiento 6(Fig. 28) que corresponde a la siguiente configuración:

tratamiento	capa oculta 1	capa salida	respuesta
	sigmoidal	lineal	Q^2
0	3	1	0.975
1	5	1	0.975
2	10	1	0.973
3	15	1	0.971
4	20	1	0.976
5	23	1	0.974
6	**25**	**1**	**0.977**
7	27	1	0.976
8	30	1	0.969

Tabla 7 *Valores de Q^2 red 5.3.2*

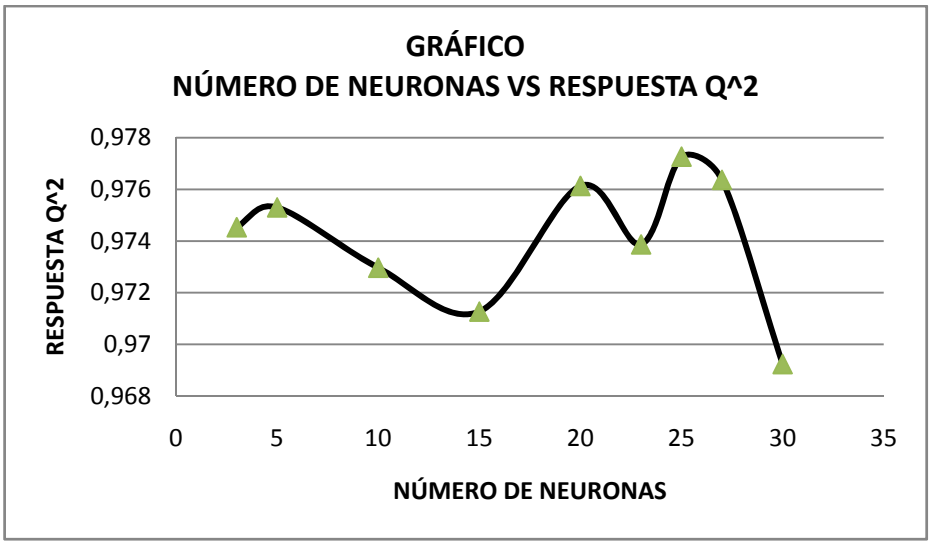

Figura 28 *Respuesta Q^2 vs número de neuronas red 5.3.2*

5.3.3 Red dos capas ocultas sigmoidales y capa de salida lineal

La tercera arquitectura posee dos capas ocultas con función de activación sigmoidal y la capa de salida con función de activación lineal. Se han corrido 9 tratamientos, es decir, 9 redes, una para cada tratamiento. Como podemos ver en la tabla 8 el parámetro que optimizamos en esta arquitectura es el número de neuronas en la capa oculta 1 y el número de neuronas en la capa oculta dos. Se ha experimentado con 10, 20, 30 neuronas tanto en la capa oculta 1 como en la 2. La capa de salida tiene 1 neurona pues se tiene un solo valor objetivo. Debido a que la etapa de entrenamiento y validación utiliza datos en forma aleatoria, esta respuesta Q^2 es promedio de tres corridas. Es decir que para generas los resultados de la tabla 5.3.3 se han corrido 3 veces cada tratamiento y luego el promedio de la respuesta Q^2 es el resultado expuesto.

Esta red presenta una respuesta promedio óptima para los tratamientos uno y dos, que corresponden a las siguientes configuraciones:

tratamiento	capa oculta 1	capa oculta 2	capa salida	respuesta
	sigmoidal	sigmodal	lineal	Q^2
1	**10**	**10**	**1**	**0.976**
2	**10**	**20**	**1**	**0.972**
3	10	30	1	0.970
4	20	10	1	0.970
5	20	20	1	0.974
6	20	30	1	0.964
7	30	10	1	0.963
8	30	20	1	0.970
9	30	30	1	0.974

Tabla 8 *Valores de Q^2 red 5.3.3*

5.3.4 Red tres capas ocultas sigmoidales y capa de salida lineal

La cuarta arquitectura posee tres capas ocultas con función de activación sigmoidal y la capa de salida con función de activación lineal. Se han corrido 27 tratamientos, es decir, 27 redes, una para cada tratamiento. Como podemos ver en la tabla 9 el parámetro que optimizamos en esta arquitectura es el número de neuronas en las capas ocultas 1, 2, 3. Se ha experimentado con 10, 20, 30 neuronas en las tres capas. La capa

de salida tiene 1 neurona pues se tiene un solo valor objetivo. Debido a que la etapa de entrenamiento y validación utiliza datos en forma aleatoria, esta respuesta Q^2 es promedio de tres corridas. Es decir que para generas los resultados de la tabla 5.3.4 se han corrido 3 veces cada tratamiento y luego el promedio de la respuesta Q^2 es el resultado expuesto.

Esta red presenta una respuesta promedio óptima para el tratamiento 12 que corresponde a la siguiente configuración:

tratamiento	capa oculta 1	capa oculta 2	capa oculta 3	capa salida	respuesta
	sigmoidal	sigmodal	sigmodal	lineal	Q^2
1	10	10	10	1	0.9746
2	10	10	20	1	0.9691
3	10	10	30	1	0.9683
4	10	20	10	1	0.9732
5	10	20	20	1	0.9725
6	10	20	30	1	0.963
7	10	30	10	1	0.969
8	10	30	20	1	0.9729
9	10	30	30	1	0.9691
10	20	10	10	1	0.9691
11	20	10	20	1	0.9721
12	**20**	**10**	**30**	**1**	**0.9793**
13	20	20	10	1	0.968
14	20	20	20	1	0.9703
15	20	20	30	1	0.9723
16	20	30	10	1	0.9725
17	20	30	20	1	0.9651
18	20	30	30	1	0.9705
19	30	10	10	1	0.9719
20	30	10	20	1	0.9702
21	30	10	30	1	0.9703
22	30	20	10	1	0.9742
23	30	20	20	1	0.9721
24	30	20	30	1	0.9583
25	30	30	10	1	0.9703
26	30	30	20	1	0.9646
27	30	30	30	1	0.9689

Tabla 9 *Valores de Q^2 red 5.3.4*

5.3.5 Determinación de la red optima

En las secciones anteriores se presentó las simulaciones con 4 arquitecturas de red.

Para cada arquitectura se ha optimizado el numero de capas utilizando como respuesta el parámetro estadístico Q^2. Se muestra en la tabla 10 un resumen de los resultados de la red óptima para cada arquitectura.

Arquitectura	**Numero óptimo de neuronas(NEU)**	**Respuesta** Q^2
1 oculta sigmoidal 1 salida sigmoidal	15 NEU capa oculta	0.978
1 oculta sigmoidal 1 salida lineal	25 NEU capa oculta	0.977
2 ocultas sigmoidal 1 salida lineal	10 NEU oculta 1 10 NEU oculta 2	0.976
3 oculta sigmoidal 1 salida lineal	20, 10, 30 NEU oculta 1, 2, 3	0.979

Tabla 10 *Valores de Q^2 para varias arquitecturas de redes neuronales*

El entrenamiento de las redes se ha realizado con datos tomados aleatoriamente, y como se puede ver en la tabla anterior la diferencia de la respuesta entre redes optimas de las diversas arquitecturas difiere en el tercer decimal, más aun se observa que las redes optimas difieren de las otras por no más que el segundo decimal, por lo cual podemos decir que la respuesta de cualquiera de las redes experimentadas es robusto frente a la variación de arquitectura y número de neuronas.

Para hacer la predicción de caudal hemos elegido la red óptima con una capa oculta de 25 neuronas con función de activación sigmoidal y capa de salida con función de activación lineal. Redes más complejas son más específicas para el grupo de datos estudiados, disminuyendo su capacidad de generalización. Respecto al tiempo necesario para entrenamiento, con la cantidad de datos utilizados (1000 datos), no existe una diferencia considerable entre las redes más complejas y las más simples, necesitando en promedio 2.3 minutos las más simples y 5.2 minutos las más complejas, en una maquina iMac 10.1 procesador Intel Core 2 Duo, velocidad del procesador 3.06, con 1 procesador, cache de nivel 2 de 3MB y 4GB de memoria.

Además la función de activación lineal en la capa de salida no tiene restricciones respecto a valores umbral, al contrario de una función sigmoidal.

5.4 Comparación de predicción con Software Comercial

Se ha calculado la predicción de caudales mediante el software Neural Machine, de Dimitri Solomatine, con el objeto de comparar resultados con nuestro programa que utiliza el Toolbox de Matlab.

Las características generales del Neural Machine son:
Número máximo de entradas: 30
Número máximo de neuronas en la capa oculta: 30

Número máximo de neuronas en la salida: 10

Número máximo de ejemplos de datos: 25000
Opciones de función de activación: sigmoidal y lineal

Debido a que Neural Machine puede tener 30 entradas como máximo, la información de entrada para entrenamiento se redujo de diez días a tres días, que para las diez estaciones completen 30 datos de entrada. Sabemos que esto disminuye la cantidad de información para el entrenamiento de la red, y que obviamente se refleja en los resultados.
Se presenta en la Fig. 29 la salida grafica de los datos de predicción generados por el prototipo (azul) y Neural Machine (rojo). La línea de resultados reales son los de la línea verde. Vale recalcar que el valor de Q^2 para esta corrida del Neural Machine es de 0.9473, inferior a los valores obtenidos mediante el prototipo en Matlab presentados en la sección 5.3.5. Por inspección visual se puede verificar que el prototipo predice mejor que Neural Machine.

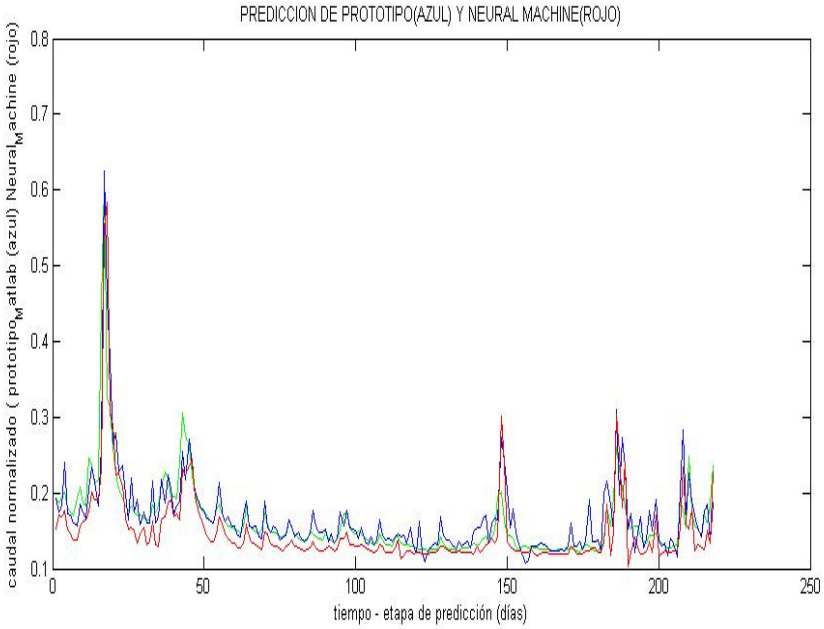

Figura 29 *Gráfico de predicción de caudales con un día de avance.*

Toolbox con dos capas ocultas sigmoidales (azul), salida Neural Machine con una capa oculta de 20 neuronas sigmoidal (rojo), y datos observados (verde).

La red utilizada para esta predicción es de una capa oculta, capa oculta sigmoidal, capa de salida sigmoidal, con 20 neuronas en la capa oculta.

5.5 Análisis de eventos extremos

En la sección anterior se determino que la red óptima es una red con una capa oculta sigmoidal y la capa de salida lineal. El numero de neuronas en la capa oculta en 25 neuronas.

Para evaluar la calidad de la predicción para caudales bajos, medios y altos se ha procedido a dividir los 218 datos de predicción en los percentiles 20 y 80. Los caudales por debajo del percentil 20 son considerados bajos, los caudales sobre el percentil 80 son considerados altos, y los caudales entre el percentil 20 y el percentil 80 son considerados caudales medios.

Comparar estadísticamente el comportamiento del modelo en estos tres grupos hemos utilizado el coeficiente de eficiente de Nash-Sutcliffe, el cual ha sido utilizado

ampliamente para evaluar modelos hidrológicos [23]. Este coeficiente varía desde -∞ hasta 1. Los valores que más se acercan a la unidad reflejan un mejor comportamiento del modelo. Este coeficiente se define como[23]:

$$E_{NS} = 1 - \frac{\sum_{i=1}^{n}(O_i - P_i)^2}{\sum_{i=1}^{n}(O_i - \bar{O})^2}$$

Donde:

O_i: Caudal observado i

P_i: Caudal calculado i

$\bar{O}$: Caudal observado promedio

En la tabla 11 se presenta los resultados del coeficiente de eficiencia para los tres grupos de caudal elegidos.

Caudales	$\mathbf{E_{NS}}$
Caudales menores al percentil 20	-12.44
Caudales mayores al percentil 80	0.50
Caudales entre el percentil 20 y 80	0.51

Tabla 11 *Valores del coeficiente de eficiencia para diversos caudales*

En las figuras 30, 31y 32 se muestra la respuesta del modelo para los caudales bajos, medios y altos.

Se puede observar que para los valores bajos del caudal el modelo tiene poca correlación con los datos observados. Este comportamiento refleja el coeficiente de eficiencia con un valor de -12.44.

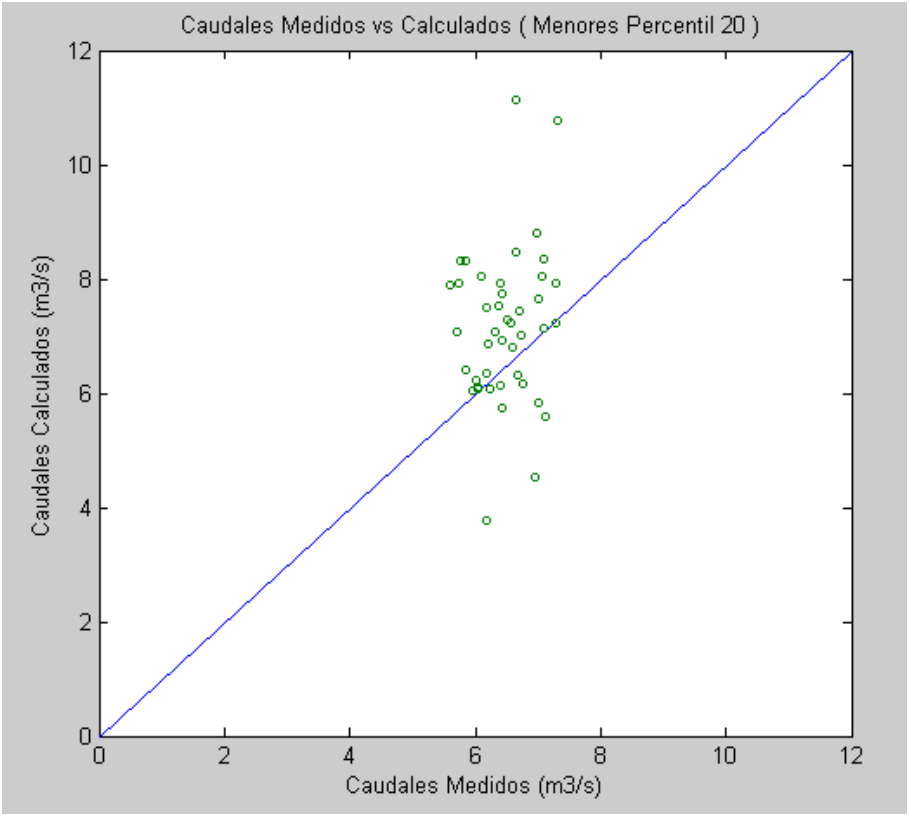

Figura 30 *Relación entre caudales observados y calculados menores al percentil 20*

Para valores de caudal alto, es decir para caudales superiores al percentil 80, el modelo se comporta mucho mejor que para caudales bajos. El coeficiente de eficiencia es de 0.50, lo cual es un reflejo de buena capacidad predicción.

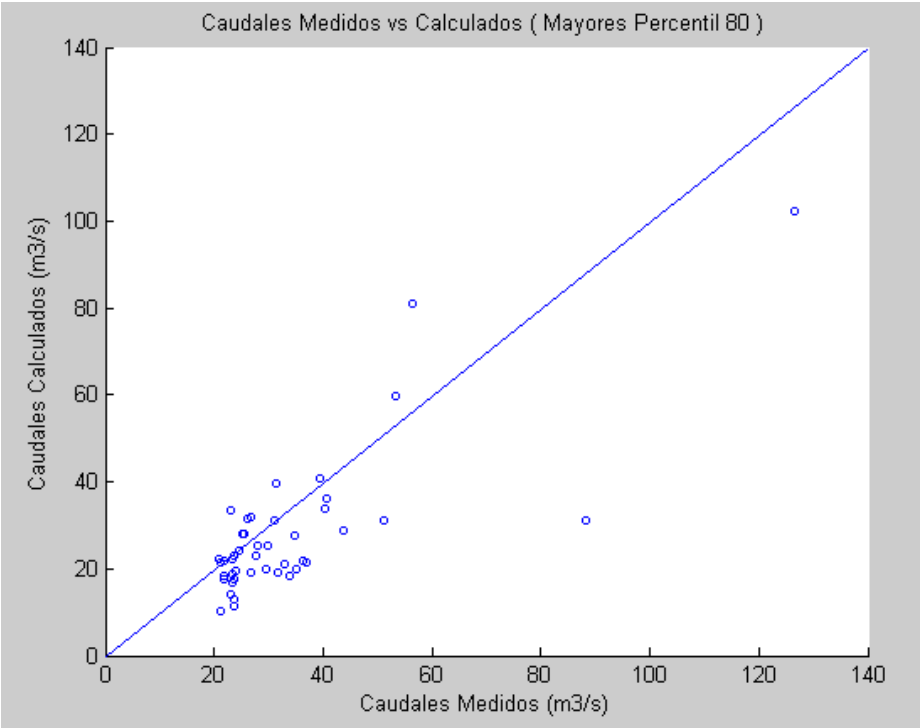

Figura 31 *Relación entre caudales observados y calculados mayores al percentil 80*

Para valores de caudal medio, es decir los caudales entre el percentil 20 y el percentil 80, el modelo se comporta bien, esto siendo reflejado por el coeficiente de eficiencia con un valor de 0.51. El buen comportamiento de predicción del modelo por lo tanto es comparable para valores de caudal alto y medio.

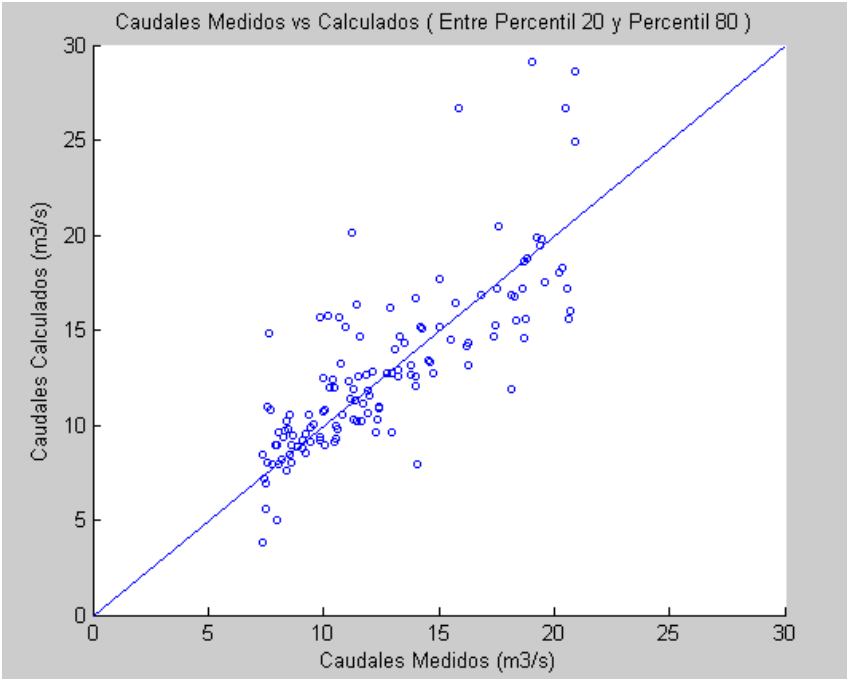

Figura 32 *Relación entre caudales observados y calculados ente el percentil 20 y 80*

5.6 Análisis de resultados

En la sección 5.1 Toolbox de Matlab se estudio la implementación de un modelo de red neuronal mediante el uso del toolbox del programa Matlab. La flexibilidad de implementación de número de capas, neuronas, funciones de activación, algoritmos de aprendizaje, y tipos de red, hacen de este programa de gran ayuda para la elaboración de un software prototipo.

El toolbox tiene la opción de clasificación de datos de entrada en tres grupos. Lo grupos de entrenamiento, validación y predicción. Una característica del entrenamiento que es destacable es el criterio del "early stopping", el cual previene del sobre entrenamiento de la red.

En la sección 5.2 Tipos de Redes Estudiadas, se ha presentado primero la implementación de una red lineal, la cual es de fácil implementación. Sin embargo este tipo de red no es adecuada para un problema altamente no lineal como el de predicción de caudales.

A continuación se ha estudiado la implementación de redes tipo multi-layer Perceptron MLP. Varios tipos se desarrollaron. La primera una red con una capa oculta sigmoidal y la de salida lineal, la siguiente una red con dos capas oculta sigmoidales y la de salida lineal, y finalmente una red con tres capas ocultas sigmoidales y la de salida lineal. Por simple inspección visual se apreció que los tres tipos de redes arrojaban buenos resultados, por lo cual se procedió a la determinación de la red óptima.

En la sección 5.3 Identificación de la Red Optima, se presento los resultados de las corridas de varias arquitecturas de red, variando en cada una de ellas el número de capas. Las arquitecturas de red estudiadas fueron: Una arquitectura con 1 capa oculta sigmoidal y la capa del salida sigmoidal, variando el número de neuronas se corrieron 8 redes. Una arquitectura con 1 capa oculta sigmoidal y la capa de salida lineal, variando el número de neuronas se corrieron 9 redes. Una arquitectura con 2 capas ocultas sigmoidales y la de salida lineal, variando el número de neuronas se corrieron 9 redes. Finalmente una arquitectura con 3 capas ocultas sigmoidales y la de salida lineal, variando el número de neuronas se corrieron 27 redes. Las óptimas de estas cuatro arquitecturas fueron comparadas, presentando una respuesta bastante similar. La red optima escogida fue la de una capa oculta sigmoidal y salida lineal con 25 neuronas en la capa oculta, por motivos de generalización debido a la simplicidad de la red, la capacidad de respuesta adecuada por tener una capa de salida lineal, y por mayor velocidad de entrenamiento respecto a redes más complejas, siendo su proporción para 1000 datos entrenados de 2.3 minutos a 5.2 minutos para las redes más complejas.

En la sección 5.4 Comparación de predicción con Software Comercial, se ha comparado nuestros resultados con los del software Neural Machine. La salida grafica en la figura 29 de la predicción de caudales de tres series de caudales, la real, la estimada con nuestro prototipo en Matlab, y la salida de Neural machine muestran mejores resultados del prototipo en Matlab. Esto puede deberse a que Neural Machine permite una entrada máxima de 30 variables, esto significa que para las diez estaciones se ingresan tan solo 3 días de historia, mientras que el prototipo recibe 10 días de historia para hacer la predicción.

En la sección 5.5 Análisis de Eventos Extremos, se divido la serie de predicción en tres grupos. El grupo de caudales bajos, que correspondes a valores inferiores al percentil 20. El grupo de caudales altos, que corresponde a caudales superiores al percentil 80. Y el grupo de caudales medios, que son los caudales entre el percentil 20 y el percentil 80. La predicción del modelo fue estudiada para cada grupo y comparada mediante el índice de eficiencia de Nash-Sutcliffe. El modelo presentó mala calidad de predicción para caudales bajos, con un valor del índice de -12.44. Para caudales medios y altos el modelo se comporto bastante bien, reflejando valores del coeficiente de eficiencia de 0.51 y 0.50 respectivamente.

CAPITULO 6

ROBUSTEZ DEL MODELO PROPUESTO

CAPÍTULO 6

ROBUSTEZ DEL MODELO PROPUESTO

6.1 Sensibilidad del modelo a las estaciones

Como se expuso en la parte teórica de esta tesis, se ha estudiado la sensibilidad del modelo a la disponibilidad en tiempo real de las variables de entrada.

Para determinar la sensibilidad del modelo a la ausencia de información de cada una de las estaciones de entrada, hemos corrido la red óptima con valores nulos de información de una estación a la vez. El error producido MSE_i es dividido para el error de la red funcionando con todas las estaciones MSE y esto se ha definido como sensibilidad; esta razón es más grande cuanto más afecte la ausencia de datos de una estación al modelo. Se presenta en la tabla 12 los resultados de esta simulación:

Simulación		1	2	3	4	5	Sensibilidad Promedio
mse todas las estaciones		0.002	0.002	0.002	0.002	0.002	
mse sin est	1	0.036	0.033	0.015	0.014	0.007	11
mse sin est	2	0.103	0.290	0.528	0.443	0.274	**159**
mse sin est	3	0.005	0.007	0.009	0.016	0.001	4
mse sin est	4	0.318	0.299	0.222	0.266	0.191	**133**
mse sin est	5	0.081	0.047	0.054	0.004	0.146	**36**
mse sin est	6	0.003	0.004	0.002	0.010	0.004	2
mse sin est	7	0.012	0.004	0.004	0.004	0.011	4
mse sin est	8	0.044	0.043	0.070	0.040	0.156	**37**
mse sin est	9	0.064	0.062	0.122	0.083	0.099	**43**

Tabla 12 *Sensibilidad promedio del modelo a las estaciones*

En la tabla 13 se muestra la sensibilidad promedio del modelo a las estaciones. Para calcular esta sensibilidad promedio se ha simulado 5 veces y para cada vez se ha obtenido los MSE correspondientes y con ellos se la sensibilidad para posteriormente promediarla.

Estación	Variables	Sensibilidad
Cancan_Soldados	Precipitación	11
Chanlud	Precipitación	159
El_Portete	Precipitación	4
El Labrado	Precipitación	133
Machangara_DJ_Chulco	Precipitación	36
Matadero_en_Sayausi	Precipitación	2
Tarqui_DJ_Cumbe	Precipitación	4
Ucubamba_en_ETAPA	Precipitación	37
Yanuncay_en_Pucan	Precipitación	43

Tabla 13 *Sensibilidad por estación*

Basados en los valores de sensibilidad las estaciones de mayor importancia para el modelo estudiado son: Chanlud, El Labrado, Yanuncay en Pucan, Ucubamba en ETAPA, Machangara DJ Chulco, reflejando menor sensibilidad para el resto de estaciones. Se ha procedido a calcular la sensibilidad de las estaciones de precipitación debido a la dependencia de la variable de caudal de días pasados con la predicción de caudal.

6.2 Análisis de Incertidumbre

Como se explicó en la sección 3.3.2, el análisis de incertidumbre estudiado en esta tesis se fundamenta en dos fuentes.

$$\sigma_Y^2 = \sum_{i=1}^{9} \sigma_{YXi}^2 + \sigma_M^2$$

La primera evalúa la incertidumbre en la predicción del caudal que el modelo genera por imprecisiones estructurales. Es decir, un modelo al ser una representación de la realidad es incapaz de capturar exactamente el fenómeno. Por lo tanto existirá una diferencia entre los valores de las variables medidas y las mismas calculadas. En este caso la predicción del caudal. Esta diferencia es entonces una deficiencia del modelo que es importante que sea evaluada, sin considerar dentro de este análisis posibles defectos o errores en los datos de entrada. Este término es σ_M^2.

Por otro lado, las variables precipitación y caudal al ser medidas están afectadas con errores de diversa índole, los cuales afectarán a la predicción del modelo. Las mejores

prácticas en la medición es cierto que limitan estos errores, pero sin embargo el error en la medición estará presente. Es por esto que el segundo término estudia la forma de propagación de los errores de las variables de entrada a través del modelo. Estos errores serán ingresados mediante distribuciones alrededor de la media. Asumimos que la distribución de los errores es normal para la generación de estos experimentos numéricos.

Se ingresa para cada estación independientemente X_i una distribución de error con varianza σ_i^2 y se obtiene como respuesta σ_{YXi}^2, que es la varianza de predicción de caudal debido a la varianza de la precipitación de la estación i. La sumatoria de estas respuestas es la incertidumbre total del modelo debido a incertidumbre en los datos de precipitación.

6.2.1 Incertidumbre del modelo en predicción

Según varios test de normalidad la distribución de los errores de predicción es normal (Ver Figura 33). Debido a que el número de datos es pequeño (número total 218 datos menos 1 outlier, 217 datos) como medida representativa de la muestra escogemos la mediana cuyo valor es –0.0189, muy cercano a cero. Procedemos a calcular la desviación estándar, su valor es 4.974, y finalmente calculamos su varianza $\sigma_M^2 = 24.7407$.

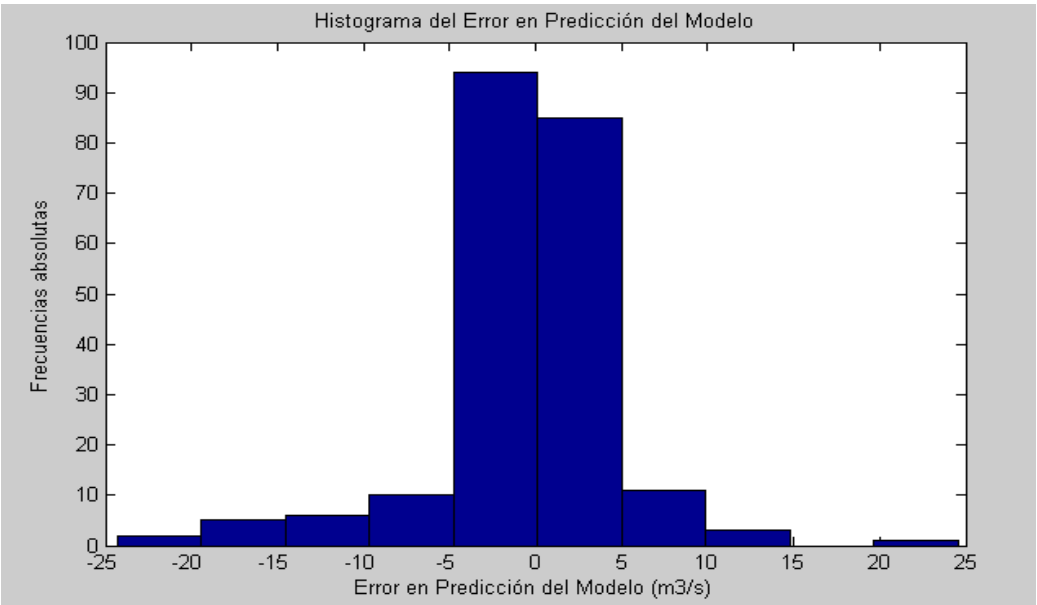

Figura 33 *Distribución del error en predicción del modelo*

6.2.2 Propagación de incertidumbre en medición

En la medición de precipitación además de los errores de lectura, apreciación y transcripción, existen los errores sistemáticos y los errores de medición por el viento [22]. Entre los factores que aportan a los errores sistemáticos son los errores por adhesión, por condensación, por evaporación y por derramamiento. Estos errores representan alrededor del 5% de la lectura. Se reporta en la literatura que los errores por efectos del viento representan hasta un 30% de la lectura de precipitación [22]. Para abordar este problema se deberían hacer estudios para estimar las correcciones de las lecturas. En esta tesis, se hace una aproximación diferente. Se ha cuantificado el efecto de los errores de medición de precipitación en la predicción del caudal del modelo.

Para estimar la propagación de la incertidumbre en los datos de precipitación del modelo se ha procedido a fijar el valor medio de cada estación, y aplicar una perturbación a este valor estación por estación, para luego calcular la perturbación en la salida que produce esta perturbación en la entrada.

Para aplicar la perturbación en la entrada se ha ingresado una distribución alrededor del valor medio de la estación (Fig. 34). Esta distribución simula la distribución del error en la medición de las precipitaciones. No se ha calculado para caudales a causa de la dependencia del caudal de predicción con valores de días anteriores como variables de entrada.

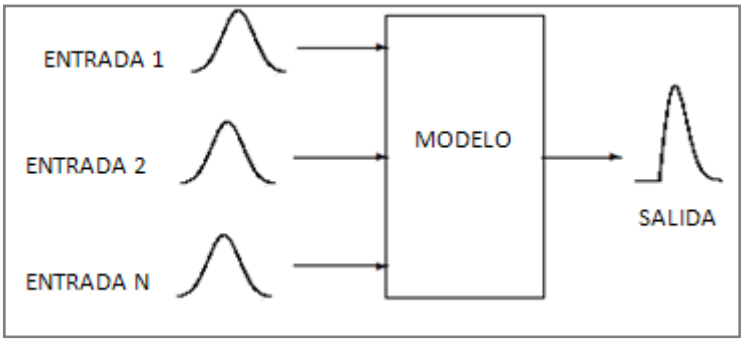

Figura 34 *Esquema de propagación de los errores de medición de precipitación*

Apoyados en la normalidad de los errores de predicción hemos simulado con una distribución normal los errores de medición en precipitación. Además, la desviación estándar se ha variando para ver el efecto de su variación en la distribución de respuesta. Las desviaciones estándar de 0.2mm, 0.4mm, 0.6mm y 0.8mm se han estudiado. En la figura 35 se puede ver la propagación del error de medición de precipitación por el modelo.

Los resultados se muestran en la tabla 14.

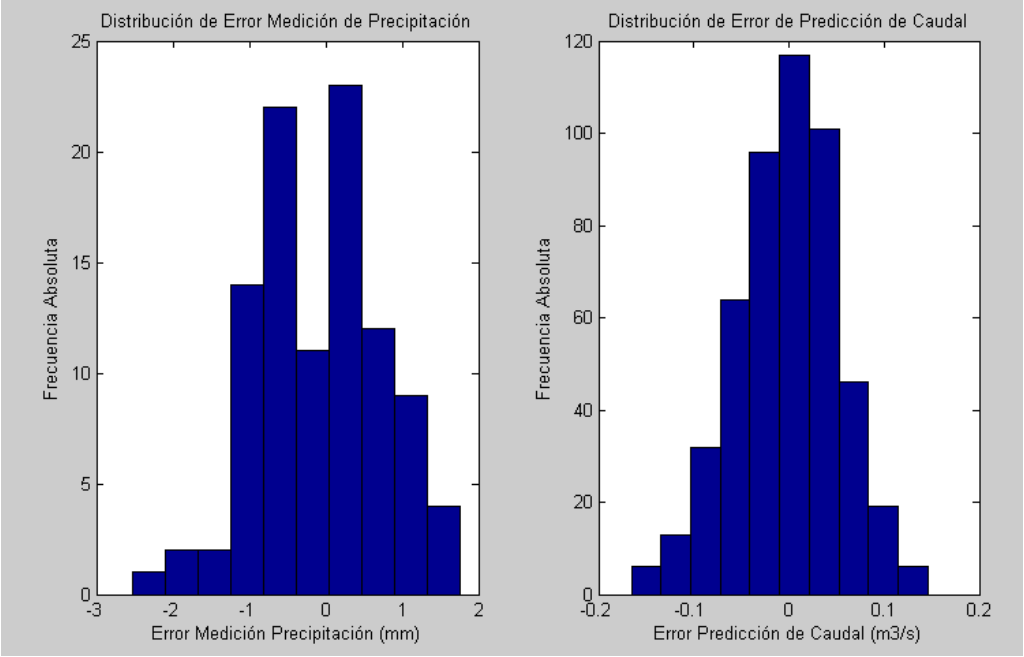

Figura 35 *Propagación de la incertidumbre de la medición de precipitación*

500 simulaciones				
Estación	**Varianza Precipitación σ^2_{Xi}**	**Varianza Predicción Caudal σ^2_{YXi}**	**Varianza Promedio Caudal**	**Coeficiente a de ajuste y= ax + b**
Estación 1	0.04	0		
	0.16	0.0001		
	0.36	0.0003		
	0.64	**0.0005**	0.00	0.000
Estación 2	0.04	0.0112		
	0.16	0.0462		
	0.36	0.1132		
	0.64	**0.1917**	**0.09**	**0.303**
Estación 3	0.04	0.0004	0.00	0.008

	0.16	0.0016		
	0.36	0.0033		
	0.64	**0.0054**		
Estación 4	0.04	0.0342		
	0.16	0.1212		
	0.36	0.2892		
	0.64	**0.5705**	**0.25**	**0.898**
Estación 5	0.04	0.0002		
	0.16	0.0008		
	0.36	0.0019		
	0.64	**0.0037**	0.00	0.005
Estación 6	0.04	0.001		
	0.16	0.0033		
	0.36	0.0078		
	0.64	**0.0153**	0.01	0.024
Estación 7	0.04	0.0113		
	0.16	0.0517		
	0.36	0.094		
	0.64	**0.1962**	**0.09**	**0.301**
Estación 8	0.04	0.0033		
	0.16	0.0165		
	0.36	0.0322		
	0.64	**0.0566**	0.03	0.087
Estación 9	0.04	0.0001		
	0.16	0.0004		
	0.36	0.001		
	0.64	**0.0016**	0.00	0.002

Tabla 14 *Varianza del error de predicción de caudal para diversas varianzas del error de precipitación*

En la tabla anterior podemos ver que la propagación de la incertidumbre de entrada depende de cada estación. Las estaciones 2,4 y 7 tienen los valores más significativos.

En la estación 2 por ejemplo para una desviación estándar de 0.8mm en la medición de precipitación, la varianza de la respuesta es 0.1917, al contrario que la estación 1 que con la misma desviación estándar propaga una varianza de 0.0005.

La estación que mayor varianza propaga es la estación 4, con una valor de 0.5705 para una desviación estándar de 0.8mm en la precipitación.

Es importante observar que las estaciones de mayor sensibilidad como la 2 y la 4 generan una mayor propagación del error de precipitación.

Otro análisis importante es la relación entre la varianza de precipitación y la varianza de caudal. Se ha detectado una relación lineal entre las dos varianzas (Fig. 36). Podemos ver en la tabla anterior que la relación más fuerte existe para la estación 4.

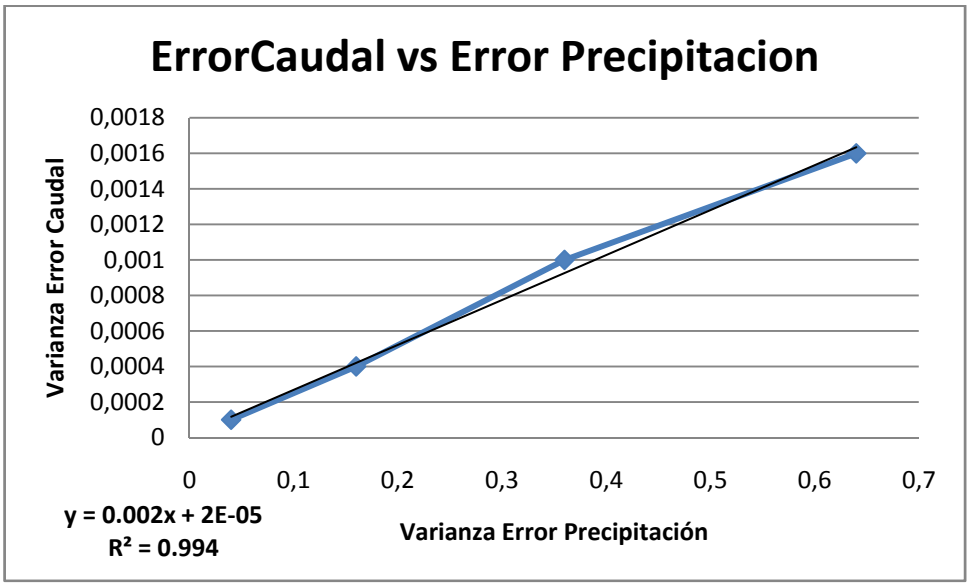

Figura 36 *Relación lineal entre la varianza del error de predicción de caudales y la Varianza del error de Precipitación*

Con estas dos fuentes de incertidumbre, la del modelo y las de propagación de errores de medición, podemos calcular la incertidumbre del modelo en predicción. Debido a que la propagación de incertidumbre depende de la varianza de ingreso en la precipitación, los valores de varianza de propagación se presentarán para una varianza de 0.64 en las estaciones de entrada.

La incertidumbre del modelo es preponderante sobre la incertidumbre de propagación de error en los datos, esto se puede apreciar en la tabla 15.

500 simulaciones			
Estación	**Incertidumbre Precipitación**	**Incertidumbre del Modelo**	**Incertidumbre de la Respuesta**
Estación 1	0.64	24.74	0.00
Estación 2	0.64	24.74	0.19
Estación 3	0.64	24.74	0.01
Estación 4	0.64	24.74	0.57
Estación 5	0.64	24.74	0.00
Estación 6	0.64	24.74	0.02
Estación 7	0.64	24.74	0.20
Estación 8	0.64	24.74	0.06
Estación 9	0.64	24.74	0.00
		Incertidumbre por los datos=	1.04

Tabla 15 *Incertidumbre total del modelo en predicción*

Podemos calcular ahora la la incertidumbre total del modelo. La incertidumbre por error estructural del modelo es $\sigma_M^2 = 24.7407$. por otro lado la incertidumbre por los datos suma 1.04. Por lo tanto la incertidumbre total es:

$$\sigma_Y^2 = \sum_{i=1}^{9} \sigma_{YXi}^2 + \sigma_M^2 = 24.74 + 1.04 = 25.78$$

6.3 Análisis de Resultados

En la sección 6.1 Sensibilidad del modelo a las estaciones, se determino la sensibilidad del modelo a las estaciones de entrada. Debido a que este modelo está orientado a la predicción de caudales en tiempo real, es importante garantizar la disponibilidad de datos para que alimenten al modelo. Sin embargo, por las condiciones operativas actuales es usual la falta de transmisión de datos de algunas estaciones por diversas circunstancias. Por lo tanto se simula la falta de información estación por estación para evaluar las que producen mayor impacto en la predicción del caudal. Las estaciones a las cuales el modelo tiene mayor sensibilidad cuando faltan sus entradas son: Chanlud, El Labrado, Machangara DJ Chulco, Ucubamba en ETAPA, Yanuncay en Pucan.

En la sección 6.2 Análisis de incertidumbre, se estudia la incertidumbre del modelo debido a dos fuentes. La primera es la incertidumbre estructural del modelo. Esto se relaciona con incapacidad de predecir que el modelo tiene en la etapa de predicción. La incertidumbre esta se ha calculado como la varianza de los errores. El valor de incertidumbre del modelo es $\sigma_M^2 = 24.7407$.

La segunda fuente de incertidumbre estudiada es la incertidumbre inducida en el modelo debido a la incertidumbre en los datos de entrada. Los datos de entrada estudiados fueron las precipitaciones. La incertidumbre de respuesta se ha observado que depende tanto de la estación, como de la incertidumbre de entrada. Consistentemente se observa una mayor propagación de incertidumbre en las estaciones más sensibles. El mayor valor de incertidumbre se ha producido para la estación 4, cuyo valor es $\sigma_4^2 = 0.5705$, correspondiente a una varianza de entrada de 0.64. La incertidumbre propagada total debido a la incertidumbre en los datos es $\sum_{i=1}^{9} \sigma_{YXi}^2$=1.04.

Calculados los dos valores de incertidumbre $\sigma_M^2 = 24.7407$ y $\sum_{i=1}^{9} \sigma_{YXi}^2$=1.04, se ha calculado la incertidumbre total $\sigma_Y^2 = 25.78$, la cual es en un 96% debido a la incertidumbre estructural del modelo.

CAPITULO 7

CONCLUSIONES Y TRABAJO A FUTURO

CAPÍTULO 7

CONCLUSIONES Y TRABAJO A FUTURO

7.1 Recapitulación

Esta tesis, "Análisis de Robustez de un Modelo de Redes Neuronales para la Predicción de Caudales en la cuenca alta del río Paute" se ha divido en dos partes. Una teórica y una básicamente aplicada. La parte aplicada corresponde a los capítulos 4, 5 y 6, de los cuales se han obtenido las conclusiones fundamentales de este trabajo.

En el capítulo 4 se estudió el tratamiento previo de los datos para alimentar el modelo de redes neuronales. Primeramente la disponibilidad de los datos y los requerimientos de datos que el modelo necesita, por ser de naturaleza distribuido. Las series de tiempo concurrentes más largas disponibles son de 1228 días, los cuales representan a registros no continuos desde el 2 de enero del 2002 hasta el 1 de agosto del 2003 para el primer intervalo y desde el 20 de marzo del 2008 hasta el 30 de diciembre del 2009.

Luego se procedió a estudiar la escala a la que los datos deben ser transformados. Utilizar la escala adecuada permitió el uso de funciones de activación tipo sigmoidales, mejoró las velocidades de entrenamiento, y aportó a la disminución de errores numéricos.

Para identificar las variables de entrada al modelo se utilizaron procedimientos estadísticos como auto-correlación y correlación cruzada entre datos de precipitación y caudal. Un aporte de datos de 10 días anteriores tanto en precipitación y caudal al de predicción se determinó como adecuado bajo criterios estadísticos y de costo computacional, cuantificado este por el tiempo de cálculo para entrenamiento de los modelos.

Finalmente en este capítulo se estudió las componentes principales de los datos para ingreso a la red. Una varianza acumulada del 88.36% se obtuvo con 6 de 10 componentes, los cual reduciría notablemente el costo computacional. Sin embargo, no se ha procedió a la implementación del modelo mediante el uso de las componentes principales por condiciones operativas de funcionamiento en tiempo real, necesitando registros más prolongados para identificar plenamente las direcciones principales.

El capitulo 5 se orientó a la determinación de la red optima para el modelo y análisis de resultados de predicción de caudales. Para esto se implementó varias arquitecturas, luego se identificó mediante parámetros estadísticos la red óptima para el problema en estudio. Una red tipo Multilayer Perceptron con una capa oculta de 25 neuronas con función de activación sigmoidal y una capa de salida con 1 neurona con función de activación lineal presentó el mejor desempeño en cuanto a correlación, mejor capacidad de generalización debido a su simplicidad y mayor velocidad de entrenamiento. Estas redes estudiadas se implementaron en el Toolbox de Matlab, lo que facilitó enormemente la experimentación de varias arquitecturas.

Posteriormente los resultados de la red óptima se compararon con los resultados del software comercial Neural Machine. Mejores resultados se obtuvieron del prototipo en Matlab que del Neural Machine, pensamos que fue debido a que Neural Machine permite solo 30 variables de entrada, mientras que el prototipo desarrollado en Matlab para esta tesis corre con 100 variables de entrada.

Finalmente en este capítulo se estudiaron los resultados del prototipo en Matlab por grupos de datos. El caudal fue dividido en 3 grupos. Caudales bajos, medios y altos, correspondientes a los percentiles 20, datos entre percentil 20 y 80, y datos sobre percentil 80. Los mejores resultados se obtuvieron para predicción de caudales medios y altos.

El capítulo 6 se estudió la robustez del modelo de redes neuronales. Simulaciones de caída de estaciones se efectuaron mediante el bloque de entrada de datos estación por estación. Las respuestas fueron variables. Para la caída de un grupo de estaciones la red respondió bien, mientras que para otras la predicción fue bastante pobre. Estaciones para las cuales la falta de datos produjo malos resultados se identificaron como la de más alta sensibilidad. Métodos estadísticos alternativos pueden ser utilizados para esta determinación mediante correlaciones pero esta tesis se ha enfocado a utilizar el modelo de redes neuronales para esta determinación. Las estaciones a las cuales el modelo fue más sensible son: Chanlud, El Labrado, Machangara DJ Chulco, Ucubamba en ETAPA, Yanuncay en Pucan,

En cuanto al análisis de incertidumbre se calculó para dos fuentes de incertidumbre. La primera fuente por la incertidumbre estructural del modelo. Esta fuente está relacionada con incapacidad de predecir que el modelo tiene en la etapa de predicción. El valor de incertidumbre del modelo es $\sigma_M^2 = 24.7407$.

La segunda fuente de incertidumbre estudiada es la incertidumbre inducida en el modelo debido a la incertidumbre en las estaciones de precipitación. La incertidumbre de respuesta se ha observado que depende tanto de la estación, como de la incertidumbre de entrada. Consistentemente se observa una mayor propagación de incertidumbre en las estaciones más sensibles. La suma total de la incertidumbre de todas las estaciones de precipitación debido a los datos es $\sum_{i=1}^{9} \sigma_{YXi}^2$=1.04

Calculados los dos valores de incertidumbre $\sigma_M^2 = 24.7407$ y $\sum_{i=1}^{9} \sigma_{YXi}^2$=1.04 , se ha calculado la incertidumbre total $\sigma_Y^2 = 25.78$, la cual es en un 96% debido a la incertidumbre estructural del modelo.

7.2 Conclusiones

Las conclusiones de este trabajo serán respuestas a los lineamientos trazados por las motivaciones y los objetivos, general y específicos.

La aplicación del método de las redes neuronales que hemos aplicado para la predicción de caudales con un tiempo de avance de 1 día ha resultado totalmente satisfactoria respecto a la calidad de los resultados, y a la velocidad de cálculo, como respalda la literatura y los resultados de esta tesis [25].

Se ha logrado programar una herramienta computacional para la predicción de caudales en tiempo real. El alto paralelismo en el procesamiento de información de las redes neuronales permite generar predicciones de caudal en segundos [26]. Si bien el entrenamiento de la red toma minutos, una vez entrenada y validada la red el cálculo de predicción es inmediato.

Los modelos hidrológicos conceptuales utilizan para predicción de caudales parámetros muy diversos acerca de las características de la cuenca y su clima. Su obtención es costosa y toma tiempo en obtener registros representativos. En esta tesis se ha comprobado que este modelo de redes neuronales que es un modelo basado en los datos presenta facilidades de implementación que otros modelos no poseen, y por lo tanto es una opción muy válida para modelar fenómenos. Sin embargo, al ser un modelo que basa su conocimiento en los datos, es imprescindible asegurar la calidad de los mismos.

Basados en el análisis de sensibilidad e incertidumbre se estudió la robustez del modelo de redes neuronales. Para las estaciones que el modelo presenta baja sensibilidad el modelo es bastante robusto ante la falta de información de las mismas. Para las estaciones que el modelo presenta alta sensibilidad se podría entrenar redes que hagan predicción sin aquellas estaciones, es decir, entrenar redes de contingencia.

Los resultados de predicción son bastante aceptables como se pudo observar en el capítulo 5. Además en la comparación con el software comercial Neural Machine el prototipo desarrollado presentó mejores resultados, identificando como causa principal la capacidad de alimentar con más variables de entrada al modelo.

La robustez del modelo frente a la incertidumbre tanto en la incertidumbre estructural del modelo como en la incertidumbre de las medidas de precipitación se ha estimado en $\sigma_Y^2 = 25.78$, es decir que la desviación estándar del error del modelo debido a estas dos fuentes es $\sigma_Y = 5.08$, lo cual tiene el orden de los caudales más bajos. Esta incertidumbre se refleja en la baja correlación que tiene el modelo justamente en el grupo de caudales menores que el percentil 20. Para los caudales medios y altos como se han definido en la capitulo 5, el modelo es robusto frente a este valor estimado incertidumbre.

Existen muchas opciones de arquitecturas de red para resolver este problema de predicción, como se mostro en el capítulo 5. Se compararon 53 arquitecturas de red y todas ellas presentaron resultados muy similares, lo cual muestra la robustez del modelo independientemente del la arquitectura utilizada. La arquitectura óptima elegida es una red tipo Multilayer-Perceptron con una capa oculta con 25 neuronas y función de activación sigmoidal. Una capa de salida con función de activación lineal, y fue entrenada mediante el algoritmo Levenberg-Marquardt. Esta red es relativamente sencilla y por lo tanto el tiempo de cómputo tanto en entrenamiento como en predicción es limitado.

Finalmente, se ha logrado utilizar como parámetros que miden la robustez del modelo, el análisis de sensibilidad y el análisis de incertidumbre. Si bien es cierto que durante la última década se ha trabajado en estos tópicos, una definición estándar de los mismos es necesaria [17]. Se ha realizado estos análisis reconociendo las condiciones de operación y la realidad de disponibilidad de datos. Por ello el análisis de sensibilidad a la disponibilidad de datos de estaciones y el análisis de incertidumbre del modelo debido a la incertidumbre del modelo y de las incertidumbres en medidas de precipitación han sido parámetros adecuados que reflejan la robustez del modelo para operación en tiempo real.

7.3 Trabajo a futuro

Si bien es cierto hemos delimitado los estudios de esta tesis con el fin de que se cumplan con los lineamientos generales propuestos en la misma, sin embargo se han encontrado temas que ameritan mayor desarrollo con el fin de que se solucionen de mejor manera varios aspectos.

Como se observó en la sección 5.4 los resultados del software Neural Machine presentó resultados menos favorables que el prototipo que se ha desarrollado en Matlab, debido a un número menor de días de historia que alimentan su red. Por ello

sería importante estudiar el efecto en la calidad de predicción con un número mayor de días de historia que los 10 que se ha utilizado en esta tesis.

Relacionado también con las variables de entrada al modelo de redes, está el estudio de las componentes principales de las estaciones. Si bien es cierto que este método exigiría la misma cantidad de estaciones para calcular las componentes principales, sin embargo luego de la elección del número de componentes principales que representen una varianza acumulada representativa de alrededor del 90%, se podría alimentar al modelo con menor variables de mayor calidad, permitiendo así mejorar los tiempos de entrenamiento y predicción.

Debido a pobre calidad de predicción del modelo para caudales bajos es importante el estudio de la calidad de la información obtenida de las estaciones. Los caudales bajos son del orden de la incertidumbre del modelo, por lo tanto son más vulnerables a ruido e información defectuosa utilizada. Es necesario desarrollar métodos para evaluar esta calidad y estudios para calcular su confiabilidad y posible corrección.

Relacionado con la calidad de los datos, es necesario realizar un análisis de las fuentes de error aplicado a nuestra realidad. Existe bibliografía que muestra esta problemática en otros países, sin embargo un estudio de este aspecto por parte de las instituciones que monitorean la cuenca de río Paute no se ha realizado.

En la etapa de recopilación de información se obtuvo registros de precipitación y caudal de varias instituciones. La poca continuidad temporal de la información refleja que por diversas razones logísticas estos registros no son permanentes. Para el adecuado funcionamiento en tiempo real del modelo realizado en esta tesis es necesario contar por lo menos con las estaciones más sensibles. Por lo tanto es necesaria la realización de redes de contingencia, las cuales deben estar orientadas a predicción en periodos prolongados de ausencia de ciertas estaciones, para lo cual estas redes deben ser entrenadas en estas condiciones.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

[1] - A.K. Jain, J Mao, K.M. Mohuiddin, Artificial Neural Networks: A Tutorial, IEEE, March 1996.

[2] - K.N. Dulal, K. Takeuchi and H. Ishidaira, A Framework for the Analysis of Uncertainty in the Measurement of Precipitation Data: a Case Study for Nepal, 21st Century COE Program on Integrated River Basin Management in Asian Monsoon Region,University of Yamanashi, Japan

[3] - Pierluigi Clap, Serie Storiche, Processie e Modelli Stocastici Per L'Idrologia e la Gestione delle Risorse Idriche, DITIC - POLITECNICO DI TORINO

[4] – M. Acosta, C. Zuluaga, H. Salazar, Tutorial de Redes Neuronales
http://proton.ucting.udg.mx/posgrado/cursos/idc/neuronales2/Transferencia.htm

[5] - Yu Hen Hu, Jenq-Neng Hwang, Handbook of neural network signal processing, Library of Congress Cataloging-in-Publication Data, editors,2001

[6] - G. Cybenko, Approximation by superposition of a sigmoidal function, University of Illinois, Department of Electrical and Computer Engineering, Technical Report 856, 1988.

[7] - T. Poggio, F. Girosi, Networks for approximation and learning, Proceedings of the IEEE, vol. 78, pp. 1481–1497, 1990.

[8] - A. Alcarraz, I. Ramirez, Proyecto de Fin de Carrera ClasiLiptus

[9] – G. Andreu, A. Crespo, J. Valiente, Selecting the toroidal self-organizing feature maps (TSOFM) best organized to object recognition, Neural Networks,1997, International Conference, Date: 9-12 Jun 1997, On page(s): 1341 - 1346 vol.2 Meeting Date: 09 jun 1997 - 12 jun 1997 Location: Houston, TX , USA

[10] - Genevieve Orr, Lecture Notes CS-449: Neural Networks, Willamette University

http://www.willamette.edu/~gorr/classes/cs449/intro.html

[11] - Manolis I. A. Lourakis, A Brief Description of the Levenberg-Marquardt Algorithm Implemened by levmar, Institute of Computer Science, Foundation for Research and Technology - Hellas (FORTH), Vassilika Vouton, P.O. Box 1385, GR 711 10, Heraklion, Crete, GREECE, February 11, 2005

[12] – S. Seung, Multilayer Perceptrons and Backpropagation learning, 9.641 Lecture 4, MIT, September 17, 2002

[13] - A. Zell et Al. Stuttgart Neural Network Simulator, User Manual Version 4.1
http://www.csc.kth.se/~orre/snns-manual/UserManual/node241.html

[14] – J. Shewchuk, An Introduction to the Conjugate Gradient MethodWithout the Agonizing Pain, Edition 11, School of Computer Science, Carnegie Mellon University, August 4, 1994

[15] - T. Lenhart et al, Comparison of two different approaches of sensitivity analysis, Physics and Chemistry of the Earth 27 (2002) 645–654

[16] - Sven Erik Magnusson, Uncertainty analysis: Identification, Quantification and Propagation, This report is financed by BRANDFORSK, Lund May 1997

[17] – L. Timbe, River flooding analysis using quasi-2d hydraulic modeling and geospatial data, PhD tesis, Universidad Catolica de Leuven

[18] - M. P. Rajurkar, Artificial neural networks for daily rainfall-runoff modeling, Hydrologkal Sciences-Jo umai-des Sciences Hydrologiques, 47(6) December 2002

[19] – G. Quinn, M. J. Melbourne, Experimental Design and Data Analysis for Biologists, Cambridge University Press, 2002

[20] - V. J. Easton, J. H. Mc Coll, Statistics Glossary, V1.1
http://www.stats.gla.ac.uk/steps/glossary/time_series.html

[21] - J. Wu et al. Scaling and uncertainty analysis in ecology: methods and applications, Edit: Springer, 2006.

[22] - K. R. Rasmussen, C. Halgreen, Some Errors in Precipitation Measurements, Nordic Hydrology, 9, Aarhus University, Denmark, 1978, 145-160

[23] – B. B. Ashagre, Swat to Identify Watershed Management Options, Tesis Presented to the Faculty of the Graduate School of Cornell University in Partial Fulfillment of the Requirements for the Degree of Master of Professional Studies By August 2009

[24] - L.R. Medsker, L.C. Jain, Recurrent Neural Networks Design and applications, Edit: CRC Press, 2001

[25] - C. Dawson and R. Wilby, An artificial neural network approach to rainfall-runoff modeling, Hydrological Sciences -Journal- des Sciences Hydrologiques, February 1998.

[26] A. W. MINNS & M. J. HALL, Artificial neural networks as rainfall-runoff Models, Hydrological Sciences -Journal- des Sciences Hydrologiques, June 1996