

Universidad de Nariño
Facultad de Ciencias Exactas y Naturales
Departamento de Física



**DISCRIMINACIÓN DE SISMICIDAD VOLCÁNICA Y TECTÓNICA
EN EL OBSERVATORIO VULCANOLÓGICO Y SISMOLÓGICO DE
PASTO UTILIZANDO REDES NEURONALES ARTIFICIALES**

TRABAJO DE GRADO

Para optar al título profesional de:

Físico

JHON JAIRO RAMÍREZ ESPAÑA

San Juan de Pasto, Colombia

JUNIO 2016

Universidad de Nariño
Facultad de Ciencias Exactas y Naturales
Departamento de Física

**DISCRIMINACIÓN DE SISMICIDAD VOLCÁNICA Y TECTÓNICA
EN EL OBSERVATORIO VULCANOLÓGICO Y SISMOLÓGICO DE
PASTO UTILIZANDO REDES NEURONALES ARTIFICIALES**

JHON JAIRO RAMÍREZ ESPAÑA

TRABAJO DE GRADO

Director:

MSc.OSCAR CADENA IBARRA

Magister en Ciencias - Geofísica

San Juan de Pasto, Colombia

JUNIO 2016

©2016 - JHON JAIRO RAMÍREZ ESPAÑA

“Las ideas y conclusiones aportadas en la tesis de grado son responsabilidad exclusiva de los autores”

Artículo 1. del acuerdo No. 324 del 11 de Octubre de 1966, emanado por el Honorable
Consejo Directivo de la Universidad de Nariño.

Todos los derechos reservados.

Nota de Aceptación

OSCAR CADENA IBARRA

Director

ALVARO RUGELES PEREZ

Jurado

SANDRA SANCHEZ SIERRA

Jurado

San Juan de Pasto, JUNIO 2016



PROPOSICIÓN N° 043
(Junio 23 de 2015)

EL COMITÉ CURRICULAR Y DE INVESTIGACIÓN DEL DEPARTAMENTO DE FÍSICA
DE LA FACULTAD DE CIENCIAS EXACTAS Y NATURALES
DE LA UNIVERSIDAD DE NARIÑO

En ejercicio de sus atribuciones legales, reglamentarias y estatutarias, y

CONSIDERANDO:

- Que el día 21 del 2016 el estudiante del Programa de Física de la Universidad de Nariño **JHON JAIRO RAMIREZ ESPAÑA** sustentó y aprobó el trabajo de grado titulado "**DISCRIMINACIÓN DE SISMICIDAD VOLCÁNICA Y TECTÓNICA EN EL OBSERVATORIO VULCANOLÓGICO Y SISMOLÓGICO DE PASTO UTILIZANDO REDES NEURONALES ARTIFICIALES**".
- Que el Director del mencionado trabajo de grado es el docente **OSCAR ERNESTO CADENA IBARRA**.
- Que mediante Acuerdo No 039 de junio 20 del 2016 el Comité Curricular designó como jurados evaluadores a los docentes **ALVARO RUGELES PERÉZ** y **SANDRA SANCHEZ SIERRA**.
- Que según el concepto de los jurados en mención el trabajo realizado por **JHON JAIRO RAMIREZ ESPAÑA** supera las expectativas para un trabajo de pregrado, debido a que para su desarrollo, el estudiante tuvo que adquirir conocimientos diferentes a los que se le impartieron durante la carrera. Además de esto, los aspectos para resaltar del trabajo son los siguientes:
 - El estudiante muestra un buen conocimiento en el estudio y clasificación de los sismos de acuerdo a las fuentes que los producen.
 - El estudiante identificó y solucionó, en un primer acercamiento, un problema cotidiano que se presenta en diversos observatorios vulcanológicos y sismológicos.
 - Los resultados de este trabajo tienen aplicación inmediata en las labores de monitoreo y prevención realizados por el OVSP.
- Que los jurados por la calificación obtenida de 100/100, sugieren otorgar al trabajo de grado la distinción de tesis **LAUREADA**.
- Que el Comité Curricular y de Investigaciones avala esta solicitud y por tanto:

PROPONE AL CONSEJO DE FACULTAD DE CIENCIAS EXACTAS Y NATURALES
DE LA UNIVERSIDAD DE NARIÑO

- **PRIMERO:** Otorgar la distinción de tesis **LAUREADA** al trabajo de grado titulado "**DISCRIMINACIÓN DE SISMICIDAD VOLCÁNICA Y TECTÓNICA EN EL OBSERVATORIO VULCANOLÓGICO Y SISMOLÓGICO DE PASTO UTILIZANDO REDES NEURONALES ARTIFICIALES**" presentado por el estudiante del Programa de Física de la Universidad de Nariño **JHON JAIRO RAMIREZ ESPAÑA**, según la parte motiva de la presente Proposición.

Dado en San Juan de Pasto, a los veinte tres (23) días del mes de junio de dos mil dieciséis (2016).





Universidad de Nariño
DEPARTAMENTO DE FÍSICA

ALVARO RUGELES PEREZ
Presidente
Comité Curricular y de Investigaciones
Departamento de Física



GP-GER 112092



SG-GER 110449



OU-SG-GER 110449

Agradecimientos

A Dios , por darme la oportunidad y capacidad de estudiar esta apasionante carrera.

A mis Padres, por el apoyo incondicional, de ellos siempre resaltaré su perseverancia la cual he sabido aprender, ellos son pilares en mi vida.

Al MSc. Oscar Cadena, por la dedicación y paciencia tenida en la elaboración de esta tesis y por ser el motivador para elaborar este trabajo.

A la Universidad de Nariño por permitir ser parte de ella y así lograr ser un profesional, como también a todos los profesores que me compartieron sus valiosos conocimientos y además fueron parte importante en mi vida universitaria.

Al Servicio Geológico Colombiano por facilitarme el acceso a herramientas vitales para que este trabajo se lleve a cabo.

Por último pero no menos importante a aquellas personas que confiaron en mi y contribuyeron a que mi paso por la universidad fuese el mejor. A mi familia Andrea Tathiana Portillo y Jhon Andres Ramírez.

A todos mi mas sincera gratitud y respeto.

A mis padres:

Ana España Casanova

Nolberto Ramírez Castillo

DISCRIMINACIÓN DE SISMICIDAD VOLCÁNICA Y TECTÓNICA EN EL OBSERVATORIO VULCANOLÓGICO Y SISMOLÓGICO DE PASTO UTILIZANDO REDES NEURONALES ARTIFICIALES

Resumen

En el presente trabajo se combinó el Análisis de predicción lineal (LPC) y Redes neuronales artificiales (RNA) con el objetivo de discriminar dos tipos de sismos: sismos volcánicos (vulcano-tectónicos, tremores, largo periodo) y tectónicos (tectónicos locales). Para ello se construyó una herramienta computacional llamada DSTYV en el software MATLAB orientada a la discriminación de estas señales sísmicas. La técnica LPC cumple la función de reducir la dimensionalidad del vector que contiene los datos de la forma de onda. Para la fase de clasificación se utilizaron las RNA tipo perceptrón multicapa (MLP) con una capa oculta. Para la elección de la topología adecuada no se recurrió a ninguna ley, sino que se probó mediante el método de ensayo-error. La aplicación de este método mostró resultados satisfactorios, la mayoría superaron el 90 %, incluso, hay una RNA de 100 % de acierto en la discriminación de datos distintos a los del entrenamiento.

Palabras claves: *Sismos Tectónicos, sismos Volcánicos, predicción Lineal, redes neuronales artificiales, perceptrón multicapa.*

DISCRIMINATION IN SEISMIC VOLCANIC AND TECTONIC IN THE OBSERVATORIO VULCANOLÓGICO Y SISMOLÓGICO DE PASTO USING ARTIFICIAL NEURAL NETWORKS

Abstract

In the present work the Lineal Prediction Coding (LPC) and artificial neural networks (ANN) is combined in order to discriminate two types of earthquakes: volcanic earthquakes (Volcano-tectonic, tremors, long period) and tectonic (local tectonics). For this built a computational tool called DSTYV in the MATLAB software oriented discrimination of these seismic signals. The LPC technique serves to reducing the dimensionality of the vector containing the data waveform. For the qualifying phase RNA multilayer perceptron (MLP) type were used with a layer hidden. For choosing the right topology no recourse to any law, but It tested by the method of trial and error. The application of this method showed results satisfactory, most exceeded 90 %, even RNA is a 100 % accuracy in discriminating different data than training.

Keywords: *Volcanic seismicity, tectonic seismicity, lineal prediction coding, artificial neural network, multilayer perceptron.*

Contenido

Título	I
Aceptación	IV
Agradecimientos	V
Dedicatoria	VI
Resumen	VII
Abstract	VIII
Índice de Figuras	X
Índice de Tablas	XI
Glosario	XII
1. Introducción	1
1.1. Red Sismológica del Volcán Galeras	2
1.2. Antecedentes	4
2. Definición del Problema	6
2.1. Planteamiento del Problema	6
2.2. Formulación del Problema	6
2.3. Objetivos	6
2.3.1. Objetivo General	6
2.3.2. Objetivos Específicos	6
2.4. Justificación	7
3. Marco teórico	8
3.1. Tipos de sismos	8
3.1.1. Clases de sismos en el VG	8
3.1.2. Sismos Tipo Tectónicos	16
3.2. Coeficientes de Predicción Lineal (LPC)	18
3.2.1. Modelo de Análisis por LPC	18
3.3. Redes Neuronales Artificiales (RNA)	22
3.3.1. La Neurona Artificial	23
3.3.2. Arquitectura de las Redes Neuronales	25
3.3.3. Definición de Red Neuronal Artificial	26
3.3.4. Operación de las Redes Neuronales Artificiales	27
3.3.5. Capacidad de Generalización	31
3.3.6. Validación cruzada	32
3.3.7. Perceptrón Multicapa	33
3.3.8. Numero de ejemplos de entrenamiento	35

4. Metodología	37
4.1. Discriminación de sismicidad	37
4.2. LPC para la representación de las señales	39
4.3. Creación del algoritmo en MATLAB e implementación del programa <i>DSTYV</i>	40
4.4. Selección de topologías para la adecuada discriminación	50
5. Resultados	51
5.1. Representación de las señales	51
5.1.1. Escogencia del orden de los LPC	51
5.1.2. Duración de los segmentos de traza	52
5.2. Elección de las mejores topologías	53
5.3. Ejecución de las RNA	56
6. Discusión	59
7. Conclusiones y Recomendaciones	61
7.1. Conclusiones	61
7.2. Recomendaciones	62
ANEXOS	63
A. Código del programa	63
Bibliografía	76

Índice de Figuras

1.1. Red Sismológica del VG (Fuente: OVSP)	3
3.1. Sismograma, espectro y espectrograma de un evento tipo VT-A en la de la estación Anganoy.	10
3.2. Sismograma, espectrograma y espectro de un evento tipo VT-B en la de la estación Cufiño (Tomado de Cadena 2011.)	11
3.3. Sismograma, espectro y espectrograma de un evento tipo LP en la de la estación Anganoy.	12
3.4. Sismograma, espectro y espectrograma de un evento tipo tornillo en la de la estación Anganoy.	13
3.5. Sismograma, espectrograma y espectro de un evento tipo Híbrido en la de la estación Urcunina.	14
3.6. Sismograma, espectro y espectrograma de un evento tipo Tremor en la de la estación Anganoy.	15
3.7. Mapa de Placas tectónicas y sus desplazamientos.	17
3.8. Sismograma, espectro y espectrograma de un evento tipo tectónico local en la de la estación Cobanegra.	18
3.9. Flujo de la señal desde su excitación hasta su salida.	19
3.10. Elementos involucrados en la extracción de los LPC	19
3.11. Sistema con filtro $A(z)$	20
3.12. Sistema con filtro $1/A(z)$	21
3.13. Neurona Artificial	23
3.14. Red Neuronal Artificial	27
3.15. Esquema de descenso por el gradiente para una función de error. La esfera azul indica el punto donde se evalúa el error y la flecha la dirección hacia donde se debe mover la esfera para alcanzar un mínimo de la función (Tomado de JOONE, 2007).	29
3.16. Errores Global y de generalización en un problema ideal, la flecha verde indica el momento en que el aprendizaje es óptimo (Tomado de CADENA, 2011)[14].	33
3.17. Representación gráfica de la función escalón o Heaviside.	33
3.18. : Funciones de activación utilizadas en las neuronas del MLP. (a) a la izquierda la correspondiente a la función log-sigmoidea y a la derecha la función tangente-sigmoidea (Tomado de CADENA, 2011).	34
4.1. Sismograma y espectro muestreado a 100 Hz.	38

4.2.	Ventana Hamming para 300 muestras.	39
4.3.	(Arriba) se muestra un tramo de evento sísmico sin ventanear y (abajo) el mismo segmento ventaneado con la función de Hamming.	39
4.4.	Estructura del programa <i>DSTYV</i>	41
4.5.	Formato de los archivos ASCII que contienen las trazas utilizadas	42
4.6.	Sección de selección de trazas	42
4.7.	Sección de selección de tiempo de corte de las trazas	43
4.8.	Sección de generación de matriz sin LPC	43
4.9.	Sección de generación de matriz de LPC	44
4.10.	Sección de selección de matriz de entrada para entrenamiento	44
4.11.	Sección de selección de matriz de objetivos para entrenamiento	44
4.12.	Sección de parámetros de entrenamiento de red	45
4.13.	Sección de entrenamiento de red	46
4.14.	Estructura de red generada mediante <i>DSTYV</i>	46
4.15.	Gráfica de los diferentes errores de entrenamiento, indicando que la mejor validación ocurrió en la época numero 10.	46
4.16.	Sección de gráficos de estado de entrenamiento	47
4.17.	Sección de almacenamiento de red	47
4.18.	Sección de ejecución de RNA	48
4.19.	Botón para ver salidas de la RNA.	48
4.20.	Sección de estadísticas de las salidas de RNA.	49
4.21.	Barra de estado <i>DSTYV</i>	49
5.1.	En La gráfica se muestran los tres ordenes LPC seleccionados en este trabajo. se puede observar que de los tres ordenes, el que mejores resultados proporciona es el de orden 7.	52
5.2.	En esta gráfica se muestra el comportamiento de los mínimos errores globales y de generalización para las 40 RNA mas destacadas. Las dos primeras topologías correspondientes a las RNA RED11277 (Figura 5.3) y RED6277 alcanzaron los valores mínimos en el error de generalización y tienen en común un momentum de 0,4.	55
5.3.	Variaciones del error global y de generalización con el número de épocas para el entrenamiento con RNA de topología 63-13-1.En esta gráfica se observa que el error global desciende hasta el final de las épocas, sin embargo el error de generalización empieza a incrementar.	56
5.4.	Porcentajes de acierto vs cada RNA utilizada en el proceso de ejecución. Las RNA con LPC 7 dan los mejores resultados para discriminar entre sismicidad tectónica y volcánica.	58
6.1.	Topología de la mejor RNA respecto a mínimo error de generalización.	60

6.2. Topología de la mejor RNA respecto a aciertos frente a datos desconocidos.	60
A.1. Entorno gráfico DSTYV.	75

Índice de Tablas

1.1. Estaciones de la red del VG y sus principales características.	3
5.1. Configuración de matrices de entrenamiento. La cantidad de eventos tectónicos y volcánicos en cada matriz es de 50 % y 50 %.	53
5.2. Valores de los errores global y de generalización para las RNA entrenadas. Esta tabla está ordenada de manera descendente de acuerdo al error de generalización,	54
5.3. Resultados de la ejecución de las mejores 30 de las 55 RNA entrenadas para la clasificación entre los tipos volcánicos y tectónicos.	57

Glosario

- Arquitectura de red:** Es un marco para la especificación de los componentes físicos de una red y de su organización funcional y configuración.
- Época:** En la teoría de redes neuronales artificiales es una medida de la cantidad de veces que todos los vectores de entrenamiento se utilizan una vez para actualizar los pesos.
- Espectro:** Gráfica de amplitud de energía vs frecuencia.
- Espectrograma:** Gráfica de frecuencia vs tiempo que representa (en colores) la energía del contenido frecuencial de una señal según va variando ésta a lo largo del tiempo.
- Flujo piroclástico:** Mezcla de gases volcánicos calientes, materiales sólidos calientes y aire atrapado, que se mueve al nivel del suelo y resulta de ciertos tipos de erupciones volcánicas.
- Frecuencia de muestreo:** Es el número de muestras por unidad de tiempo que se toman de una señal continua para producir una serie de tiempo discreta.
- Fuente:** Lugar donde se generan sismos.
- Iteración:** Acto de repetir un proceso con la intención de alcanzar una meta deseada, objetivo o resultado..
- MATLAB:** Abreviatura de MATrix LABoratory, es una herramienta de software matemático que ofrece un entorno de desa-

rrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M).

Red neuronal artificial: Es un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso de los animales.

Retropropagación: algoritmo de aprendizaje supervisado que se usa para entrenar redes neuronales artificiales.

Sismo / Seísmo: Movimiento de la Tierra producido por el paso de las ondas generadas por el desplazamiento repentino del suelo por debajo de la superficie de la Tierra.

Sismograma: Registro hecho por un sismógrafo.

Sismómetro de Corto Periodo : Sismómetro con una respuesta en velocidad normalmente plana en el rango de frecuencia mayor a 1 Hz.

Sismómetros de Banda Ancha: Sismómetro de respuesta en velocidad con un rango de frecuencia más amplio, incluyendo frecuencias desde los 0,008 Hz.

Sismómetros Triaxiales: Sismómetros que pueden registrar la velocidad del suelo en sus tres dimensiones.

telemetría: Tecnología que permite la medición remota de magnitudes físicas y el posterior envío de la información hacia el operador del sistema.

Traza: Representación gráfica de un sismo.

Volcán andesítico: Tipo de volcán que contiene lava de composición intermedia, es decir, entre el 55 % y el 65 % en sílice.

Capítulo 1

Introducción

El Volcán Galeras (VG) está localizado en el Departamento de Nariño, aproximadamente a 9 km al occidente de la ciudad de San Juan Pasto, con una altura de 4.276 m.s.n.m. y con coordenadas $1^{\circ}13'44''$ de latitud Norte y $77^{\circ}21'33''$ de longitud Occidente. Su cono activo está ubicado a *9km* al occidente de Pasto, la capital del departamento, con 440.000 habitantes aproximadamente. El VG es un estrato-volcán conformado por depósitos de flujos de lava andesítica, flujos piroclásticos y depósitos de caída. Después del ciclo eruptivo entre 1942 y 1944, el VG mostró nuevamente actividad a finales de 1988; el Servicio Geológico Colombiano antes (INGEOMINAS) en 1989 empezó la observación permanente de su actividad con el establecimiento del Observatorio Vulcanológico y Sismológico de Pasto (OVSP)[1].

En las operaciones del OVSP la preparación de la información sismológica como insumo para el análisis e interpretación de los niveles de actividad volcánica y tectónica inicia con la generación de histogramas a partir de la clasificación de la sismicidad. Esto se lleva a cabo manualmente, analizando uno a uno los registros sísmicos, lo que implica, un trabajo largo y dificultoso y por consiguiente una gran inversión de tiempo del personal. Estas dificultades son especialmente agudas en épocas de actividad sísmica tanto volcánica como tectónica.[2]

En este trabajo se diseña una metodología basada en la combinación de Análisis de predicción lineal (LPC) y Redes neuronales artificiales (RNA) e implementar una herramienta computacional que permita discriminar entre sismos volcánicos y tectónicos. La representación de las señales por medio de LPC se justifica porque reduce la dimensión de los vectores de entrada al sistema clasificador. RNA adecuadamente entrenadas son utilizadas para la clasificación.

El primer trabajo que consistió en discriminación de sismicidad con la combinación de LPC con las RNA fue en el volcán vesubio para dar solución al problema de clasificación en sismología volcánica.[13]

En la fase de discriminación se realiza la clasificación entre dos tipos de sismicidad: volcánica , asociada a actividad en el edificio volcánico y tectónica donde el edificio volcánico no tiene incidencia. Este trabajo no tiene en cuenta segmentos de ruido que haya sido originado por factores como tormentas eléctricas, inducción de voltajes no deseados, etc. Los trabajos de clasificación de sismos para el VG de (Dimaté & Cadena 2011), contienen parte de la metodología utilizada en la presente investigación.

1.1. Red Sismológica del Volcán Galeras

La red sismológica instalada en el VG ha tenido algunas variaciones desde su despliegue inicial en el año 1989, actualmente está constituida por sensores tanto de Banda ancha (BA) como de Corto periodo (CP), se cuenta con cinco sensores triaxiales y cuatro de componente vertical. Los sensores de CP utilizados son tipo L4C y L4C-3D, mientras que los de BA son tipo Guralp de 60 y 120 segundos. Todos estos sensores son transductores de velocidad. La disposición de los instrumentos está fuertemente condicionada por la topografía de la zona (Figura 1.1 y Tabla 1.1). Las señales de los sensores de BA se digitalizan en campo y se transmiten teleméricamente a través de radios de doble vía y espectro ensanchado. La información que registran los sensores de CP es enviada a través de la banda UHF y se digitaliza en la sede del OVSP, con excepción de las estaciones Calabozo y Cráter2 cuya información se digitaliza en campo[2].

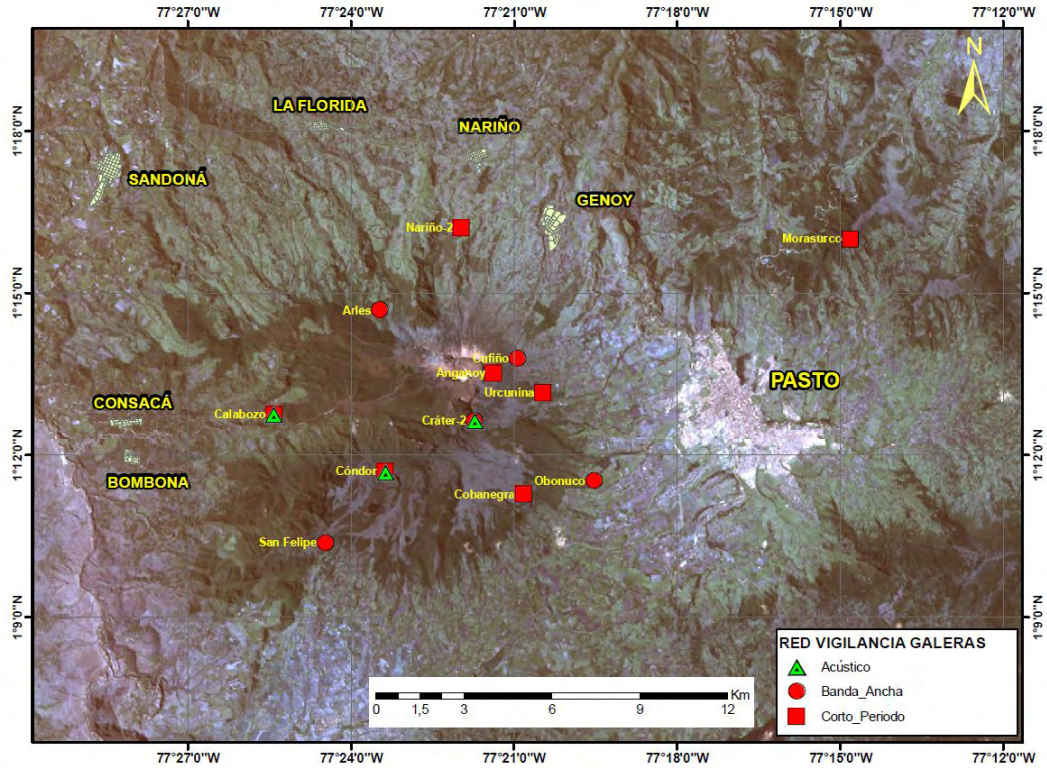


Figura 1.1: Red Sismológica del VG (Fuente: OVSP)

Estación	Tipos de datos a transmitir	Tipo de sensor	Componentes	Distancia al Cráter(Km)	Altitud (m.s.n.m)
Anganoy	Analógico	Corto Periodo	Vertical	1	4227
Cráter-2	Analógico	Corto Periodo	Vertical	1.4	4063
Cráter-2	Digital	Banda Ancha	Triaxial	1.5	4052
Cufiño	Digital	Banda Ancha	Triaxial	1.8	3828
Urcunina	Analógico	Corto Periodo	Triaxial	2.8	3494
Cobanegra	Analógico	Corto Periodo	Vertical	4.8	3625
Obonuco	Analógico	Corto Periodo	Triaxial	5.9	3021
Cóndor	Analógico	Corto Periodo	Vertical	4.4	3977
Nariño-2	Analógico	Corto Periodo	Vertical	4.8	2846
Calabozo	Digital	Corto Periodo	Triaxial	6.7	2313

Tabla 1.1: Estaciones de la red del VG y sus principales características.

1.2. Antecedentes

DISCRIMINATION OF EARTHQUAKES AND UNDERWATER EXPLOSIONS USING NEURAL NETWORKS.[13]

Los autores de este trabajo aplicado a datos en el observatorio del volcán Vesubio fueron los primeros que combinaron los LPC con las RNA para dar solución al problema de clasificación en sismología volcánica. En su trabajo se clasifican a posteriori (no en tiempo real) sismos VT y explosiones en canteras.

DISCRIMINATION OF SEISMIC SIGNALS USING ARTIFICIAL NEURONAL NETWORKS[3].

En este trabajo se hace una clasificación entre sismos locales y explosiones químicas usando RNA de datos obtenidos en el laboratorio de geofísica del Centro Nacional para las ciencia e investigación técnica (Morocco, USA). La metodología usada en este trabajo consistió en tres bloques:

- Representación
- Reducción de dimensionalidad
- clasificación

Para la reducción de dimensionalidad los autores propusieron un algoritmo basado en la proyección aleatoria y el análisis de componentes principales.

DETECCIÓN Y CLASIFICACIÓN AUTOMÁTICA DE REGISTROS SÍSMICOS EN EL OBSERVATORIO VULCANOLÓGICO Y SISMOLÓGICO DE PASTO UTILIZANDO REDES NEURONALES ARTIFICIALES[2].

Con análisis de predicción lineal(LPC) y redes neuronales artificiales (RNA) Cadena detecta y clasifica tres tipos de sismos volcánicos: volcano-tectónicos (VT) tipo A, largo periodo (LP) y tremor (TRE). La clasificación de estos sismos se hacen con una precisión por encima del 90 %.

CLASIFICACIÓN DE DATOS GEOFÍSICOS MEDIANTE EL USO DE ATRIBUTOS SÍSMICOS NO CONVENCIONALES Y REDES NEURONALES.[4]

En este trabajo se describe una metodología para la clasificación de datos geofísicos mediante el entrenamiento y uso de redes neuronales.

El método de utilización de las redes neuronales en este trabajo pertenece a la categoría de métodos de aprendizaje supervisado lo que requiere que las clases a identificar sean conocidas en un conjunto del sub-volumen de datos (datos de entrenamiento).

Capítulo 2

Definición del Problema

2.1. Planteamiento del Problema

En vista de que el proceso de discriminación se realiza de forma manual, analizando uno a uno los registros sísmicos, esto que implica un seguimiento minucioso para evitar la pérdida de información importante, y además una gran inversión de tiempo del personal. Estas dificultades son especialmente agudas en épocas de crisis cuando se registran centenares de sismos volcánicos diariamente o sismos tectónicos ocasionalmente .

Por otra parte, hay casos en los que existen opiniones diferentes sobre la clasificación de un mismo sismo.

2.2. Formulación del Problema

¿ Es posible discriminar sismos tectónicos de sismos volcánicos utilizando RNA?

2.3. Objetivos

2.3.1. Objetivo General

Discriminar automáticamente entre sismicidad volcánica y tectónica usando RNA.

2.3.2. Objetivos Específicos

- Diseñar una estrategia para la discriminación de sismos tectónicos y volcánicos.

- Pre-procesar, representar y reducir la dimension de los registros utilizando LPC.
- Diseñar y entrenar adecuadamente las RNA para la discriminación.
- Diseñar y codificar un programa para aplicar la metodología propuesta para este estudio.

2.4. Justificación

La generación de técnicas discriminación que en este caso hace el uso de RNA, son de mucha utilidad para el monitoreo de señales sísmicas, especialmente en épocas de crisis, en los que es de vital importancia tomar decisiones rápidas y efectivas para evitar un desastre natural. Un procedimiento de discriminación como el desarrollado en este trabajo permitiría ahorrar tiempo en el proceso de clasificación, sin comprometer la precisión de los resultados.

Los resultados que se den en este trabajo servirán como base para estudios posteriores en sismicidad ocurrida tanto en Nariño como en Colombia, ya que esta tesis tiene aplicación en cualquier tipo de datos sísmicos cuyos sismogramas estén interpretados en formatos similares a los utilizados en este trabajo (Formato ASCII).

Capítulo 3

Marco teórico

3.1. Tipos de sismos

Los sismos pueden clasificarse según la fuente que los origina. Este trabajo se centra en el estudio de dos tipos de sismos naturales que son los tipo tectónicos y volcánicos. Las frecuencias predominantes de un sismo permiten diferenciar una actividad sísmica tectónica o volcánica. Los sismos cuyas frecuencias predominantes son bajas $f < 3Hz$, usualmente son de actividad volcánica. Es por ello que los sismos tectónicos son de mayor impacto que los volcánicos.

3.1.1. Clases de sismos en el VG

En los volcanes del mundo se presentan sismos que tienen firmas semejantes y que se han asociado con el tipo de fuente que los origina. Por ejemplo, la fractura de material cortical en el edificio volcánico, sin que importe la causa que la produjo, genera eventos sísmicos que tienen arribos impulsivos y sus fases P y S son fácilmente reconocidas, estos sismos son llamados volcano-tectónicos, se distinguen por las siglas VT[5][11]. Puede suceder que los sismos VT sean producto de actividad volcánica o de acumulación de esfuerzos tectónicos que se transmiten hacia la región volcánica; el hecho de que su foco esté localizado en inmediaciones de la estructura volcánica es suficiente para clasificarlo como VT.

Existen eventos sísmicos volcánicos relacionados con la dinámica del sistema magma-gas e hidrotermal (Chouet, 1996). Este tipo de sismos se divide en dos grupos diferenciados por el tiempo de excitación de la fuente generatriz. Se llaman sismos de largo periodo o LP aquellos en que su fuente es transitoria y tremor o TRE cuando la fuente es persistente

en el tiempo. Chouet considera que una sucesión de LP puede ser tratada como un solo episodio de tremor. Otro tipo de sismos involucra tanto fractura de material sólido como movimiento de fluidos, estos son llamados Híbridos o HYB[6]. En algunos volcanes se observan eventos que responden a particularidades en su estructura o dinámica, por ejemplo los llamados tornillos o TO (Gómez et al., 1999) registrados en el VG y en contados volcanes del mundo. La clasificación de sismos en el VG ha sufrido modificaciones. Cuando se inició el monitoreo sismológico del volcán y teniendo en cuenta que no se contaba con una línea base de información sobre la cual referir la sismicidad, se decidió adoptar la clasificación que hasta el momento se utilizaba en el volcán Nevado del Ruiz, basada en la de (Minakami, 1974 y Latter, 1979), y modificada por (Malone, 1983), que era similar a la que se utilizaba en el volcán Santa Helena. Luego, con los estudios de (Lahr et al, 1994) y (Chouet, 1981, 1988, 1992, 1996), se adoptó la clasificación de: largo periodo, volcano-tectónico, tremor armónico y se incluyeron además el tremor espasmódico, el tornillo y el mariposa (Gómez et al., 2005).

Las definiciones que se describen a continuación se han tomado del Catálogo de señales sísmicas volcánicas de Colombia , que describe los tipos de sismos volcánicos dentro de una división superior que contiene: señales sísmicas volcánicas transitorias y señales sísmicas volcánicas continuas[23].

Sismos volcánicos de fuente transitoria

Hace referencia a señales cuya fuente actúa en un tiempo relativamente corto, generalmente unos pocos segundos, luego de esto se generan las ondas de coda del evento. Existen señales transitorias asociadas con fractura de material cortical y también con movimiento de fluidos, dentro de estas señales se encuentran los tipos: VT, LP, HYB y TO.

Sismos volcano-tectónicos (VT)

Dentro de este tipo de sismos podemos distinguir dos clases, VT-A y VT-B:

Sismos volcano-tectónicos tipo A o VT-A

En general ocurren a profundidades entre 2 y 10 km desde la cima del volcán (Figura 3.1). En su registro en el dominio del tiempo es posible distinguir con facilidad las fases P y S. El decaimiento de la envolvente de sus ondas de coda muestra una forma exponencial disminuyendo su amplitud rápidamente con el tiempo. El espectro de este tipo de eventos contiene altas frecuencias, mayores que 5 Hz; los picos que aportan con mayor energía a la señal están generalmente en una banda entre los 6 y 8 Hz y sus ondas de coda pueden contener frecuencias significativamente energéticas, incluso hasta los 15 o 20 Hz. En el campo de los desplazamientos el contenido espectral no supera los 6 Hz. La fase inicial de los sismos tipo VT es frecuentemente impulsiva, ocasionalmente se observan arribos emergentes en algunas estaciones como consecuencia de pérdidas de energía en la onda P o por la interacción de esta con material más fluido dentro del sistema volcánico (Latter, 1984), también influye el patrón de radiación de la fuente. La distribución de eventos VT en el tiempo es, por lo general, a manera de enjambre, con magnitudes aleatorias, en contraste con los eventos no volcánicos que muestran una secuencia de réplicas que siguen al evento principal y van disminuyendo su magnitud en el tiempo.

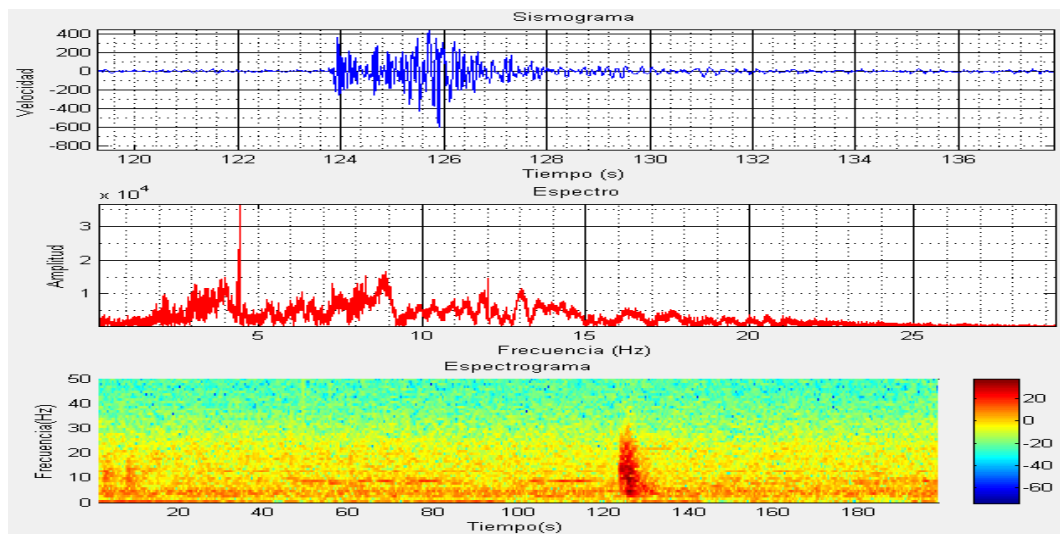


Figura 3.1: Sismograma, espectro y espectrograma de un evento tipo VT-A en la de la estación Anganoy.

Sismos volcano-tectónicos tipo B o VT-B

Este tipo de eventos se presentan generalmente a profundidades menores que 2 Km. Entre las características más destacadas de su forma de onda están sus duraciones relativamente largas, el arribo de la fase P es emergente y es bastante difícil reconocer la fase S. En su espectro (Figura 3.2) se observa que la energía del sismo está concentrada en una banda de bajas frecuencias, generalmente menores que 5 Hz, por otro lado, en sus ondas de coda es posible encontrar indicios, de dispersión de ondas superficiales. Una posible explicación es que por tratarse de focos poco profundos encuentran una gran cantidad de dispersores en su trayecto que afectan las frecuencias más altas durante la propagación de la onda.

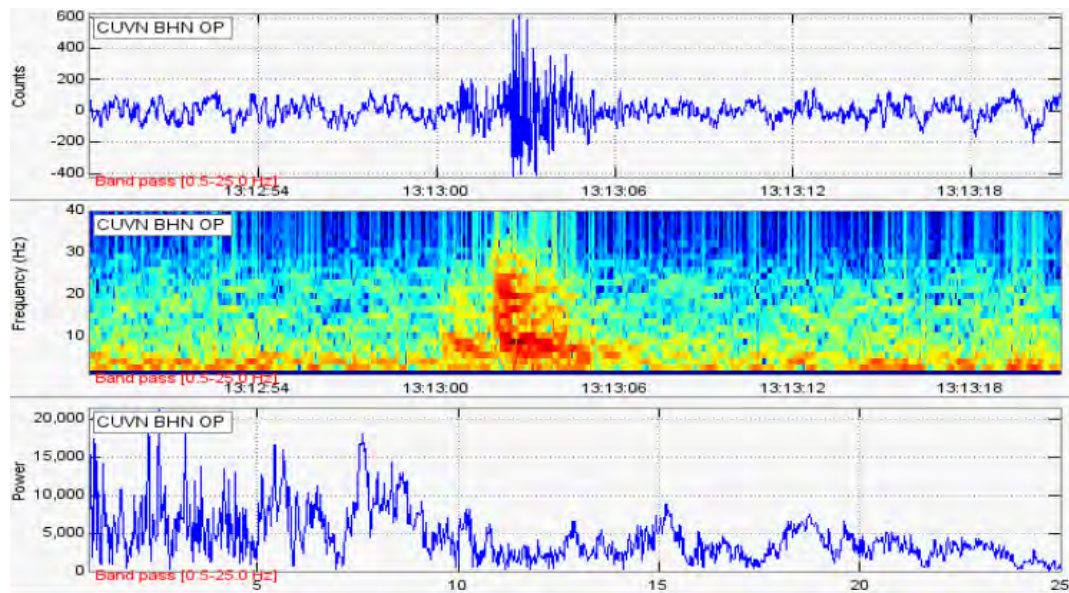


Figura 3.2: Sismograma, espectrograma y espectro de un evento tipo VT-B en la de la estación Cufiño (Tomado de Cadena 2011.)

Estos sismos, son causados por fracturamiento de material cortical cuya fuente puede consistir principalmente en un mecanismo de doble-par de fuerzas, sin embargo la diferencia con los registros de los sismos VT-A podría reflejar efectos de camino ya que al ubicarse en capas superficiales estas contienen heterogeneidades de piroclastos y lavas, incrementando la dispersión (scattering)[2].

Sismos tipo Largo periodo (LP)

Los sismos LP se caracterizan porque en su firma generalmente es posible distinguir la forma de onda de sus armónicos dominantes, que generalmente se encuentran por debajo de los 5 Hz, superpuestos a estos armónicos se encuentran componentes menos energéticas que alcanzan los 13 Hz. Cuando el foco del evento es cercano a la estación de registro se observa un arribo impulsivo, similar al de un VT-A, que en las estaciones más lejanas se observa como emergente. En los sismos LP el decaimiento de ondas de coda es relativamente lento por lo cual la duración de estos eventos es prolongada, es difícil distinguir la onda S. Las formas de onda son a veces similares a las del tipo VT-B. Su firma en ocasiones exhibe pulsos que pueden originarse por la superposición de armónicos de frecuencias cercanas o ser originados por vibraciones pulsátiles de la fuente misma (Figura 3.3).

Los sismos tipo LP en ocasiones muestran una señal emergente de bajo nivel energético, a modo de precursor, seguido a este se presenta el evento que es mucho más energético respecto a su precursor y con arribo impulsivo. Es común observar componentes de alta frecuencia al inicio de la señal.

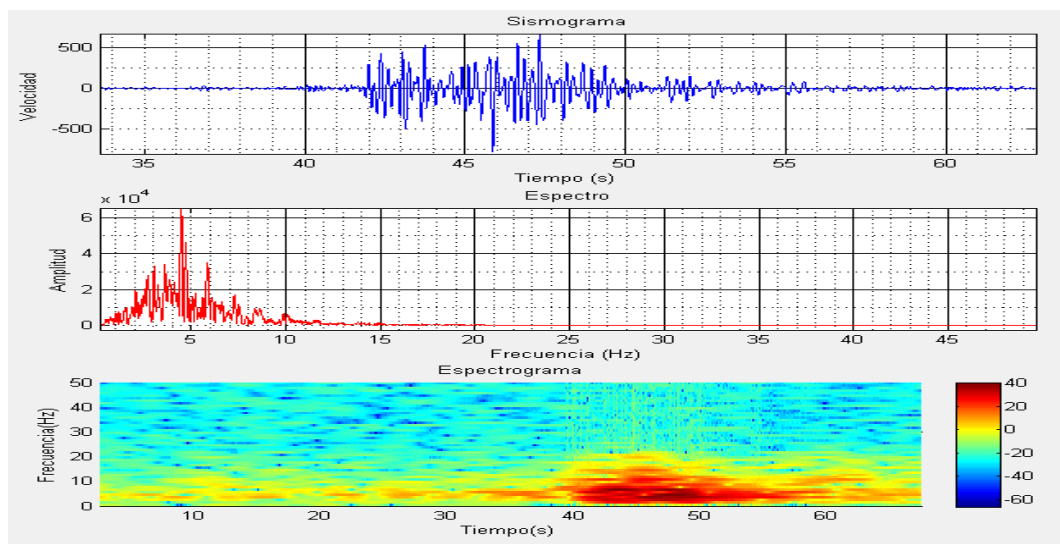


Figura 3.3: Sismograma, espectro y espectrograma de un evento tipo LP en la de la estación Anganoy.

En general, los focos de estos sismos se ubican a profundidades menores que 2 km. Según Chouet, (1985, 1988) estos sismos pueden ser causados por resonancia de conductos o fracturas llenas de fluidos e iniciados por transitorios de presión[6]. Existe una variedad de modelos para explicar la génesis de los sismos LP, estos van desde la resonancia en conductos o grietas (Chouet, 1988), hasta fenómenos de resonancia dentro del propio magma. Estos modelos explican gran parte de las características espectrales de este tipo de eventos.

Sismos tipo Tornillo (TO)

Este nombre fue acuñado en Colombia por el personal del OVSP para distinguir un tipo especial de sismos de largo periodo cuya fuente está asociada con procesos de resonancia[1]. Su firma muestra un inicio impulsivo tras el cual se observa que uno o varios armónicos alargan su coda por tiempos relativamente largos. La manera en que decae la amplitud del sismograma a lo largo del tiempo es lenta, lo que implica que la tasa de pérdida de energía en la cavidad resonante es lenta también. Su espectro muestra uno o pocos picos en las frecuencias de los armónicos que aportan la energía del sismo (Figura. 3.4).

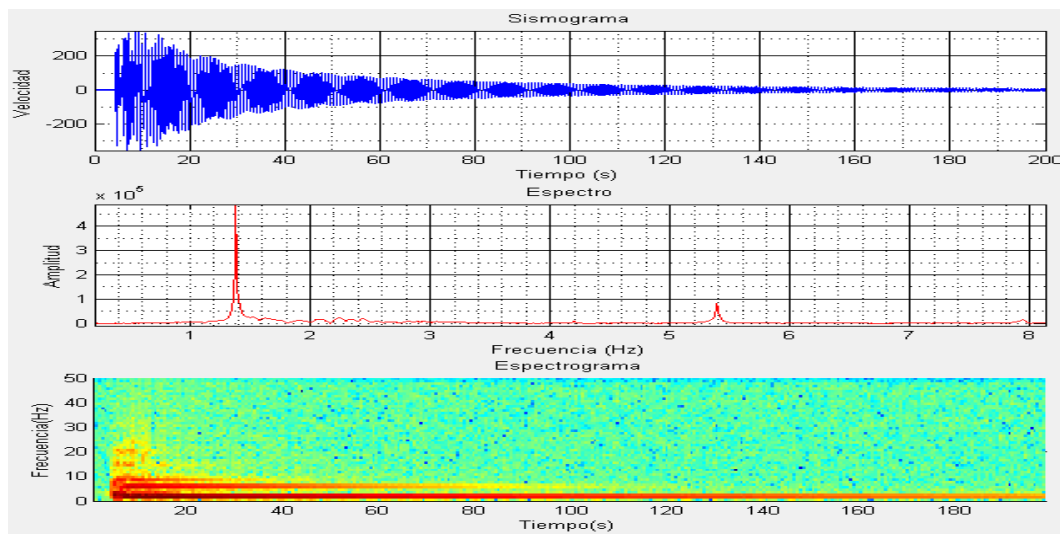


Figura 3.4: Sismograma, espectro y espectrograma de un evento tipo tornillo en la de la estación Anganyo.

En el VG generalmente se presentan en el contexto de un sistema cerrado. Un transitorio de presión dentro de una grieta, por ejemplo, podría ser el inicio de la resonancia que dependiendo de la composición y propiedades físicas de los fluidos involucrados determinarían los armónicos dominantes en su espectro y la duración del evento.

Sismos tipo Híbridos (HYB)

Muestran características originadas por la combinación de procesos que involucran movimiento de fluidos y fracturamiento de material cortical por lo cual se ha sugerido que se trata de la combinación de eventos tipo VT-A o VT-B con sismos tipo LP. Generalmente su arribo es impulsivo, dominado altas frecuencias y en muchas ocasiones es posible distinguir la fase S. Sus ondas de coda muestran patrones armónicos no dispersivos, característicos de los eventos tipo LP. Como los sismos VT, los sismos HYB muestran diferentes polaridades en sus arribos en las diferentes estaciones, mientras que los LP tienen igual polaridad en las diferentes estaciones de registro(Figura 3.5).

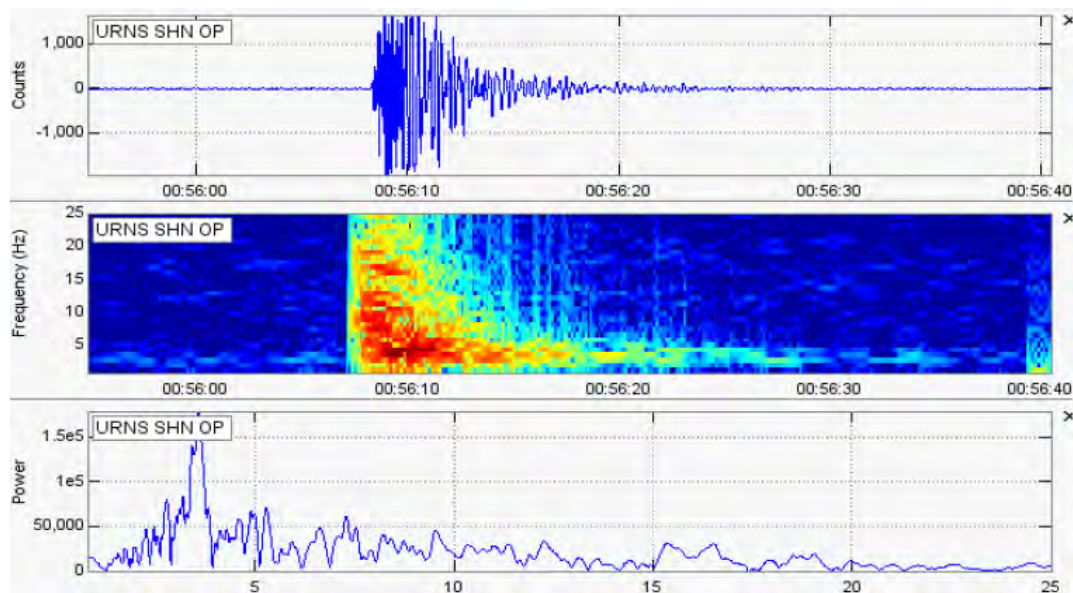


Figura 3.5: Sismograma, espectrograma y espectro de un evento tipo Híbrido en la de la estación Urcunina.

Señales sísmicas volcánicas continuas

El sismograma de las señales volcánicas continuas puede estar asociado con una fuente persistente en el tiempo relacionada con dinámica de fluidos o con procesos de deslizamiento de roca o lahares.

Tremor Volcánico (TRE)

Este tipo de sismicidad presenta una firma continua en el tiempo que refleja una excitación persistente de la fuente. El sismograma del tremor volcánico puede tener variaciones en amplitud producidas por cambios en la energía del fenómeno que lo produce, aunque en otros casos la amplitud es estable. La duración del tremor puede variar desde unos pocos minutos hasta meses. En ocasiones es posible distinguir una secuencia de sismos, de tipo LP por ejemplo, que se concatenan formando una señal continua que también es considerada como tremor (Figura 3.6). Por sus características espectrales se puede diferenciar el tremor armónico y por su secuencia temporal el tremor espasmódico.

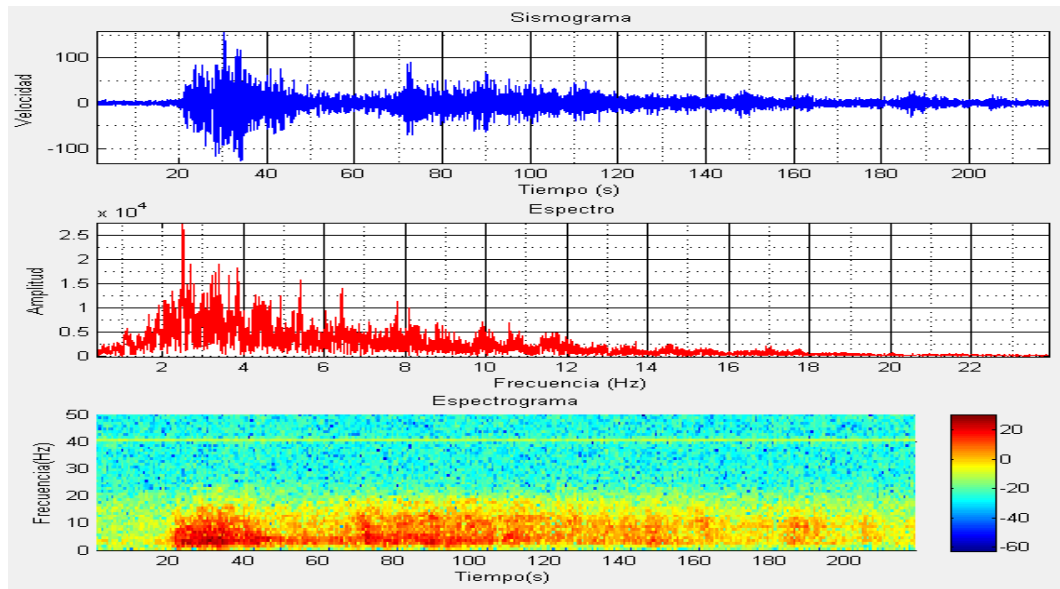


Figura 3.6: Sismograma, espectro y espectrograma de un evento tipo Tremor en la de la estación Anganoy.

No Clasificables

Sismos que por su bajo contenido energético es difícil establecer su origen y por consiguiente de incluirlos en alguno de los tipos mencionados. Sin embargo son un recurso importante en el análisis de la actividad volcánica ya que nos dan una idea de la recurrencia con la que actúan determinadas fuentes. El problema en la caracterización de este tipo de sismos radica en la dificultad para distinguirlos del ruido base.

Ruido

Para este estudio se define como ruido cualquier señal que no sea un sismo de los tipos mencionados anteriormente. De acuerdo con esto, inclusive los eventos tectónicos que se localicen fuera del edificio volcánico se consideran ruido. Entre los factores de mayor incidencia sobre el ruido base están el clima y la acción del hombre cerca de las estaciones sismológicas.

3.1.2. Sismos Tipo Tectónicos

Se generan por la actividad propia de las placas tectónicas. La corteza de la Tierra está conformada por una docena de placas de aproximadamente 70 km de grosor, cada una con diferentes características físicas y químicas. Estas placas se están acomodando en un proceso que lleva millones de años y han ido dando la forma que hoy conocemos a la superficie de nuestro planeta. Entonces una placa comienza desplazarse sobre o bajo la otra originando lentos cambios en la topografía. Pero si el desplazamiento es dificultado, comienza a acumularse una energía de tensión que en algún momento se liberará y una de las placas se moverá bruscamente contra la otra rompiéndola y liberándose entonces una cantidad variable de energía que origina el sismo, es decir que en este caso se produce el desequilibrio de las capas de la corteza terrestre producido por el fenómeno de la contracción que produce las arrugas o pliegues.

Zonas enteras de estratos pueden quedar aplastadas y desechas, las capas pueden desligarse las unas sobre las otras, dislocarse, agrietarse. Debido precisamente a estos fenómenos de dislocaciones interna, el estrato conmovido por el choque produce vibraciones, las cuales se propagan instantáneamente a todas las capas rocosas superiores y circundantes[4].

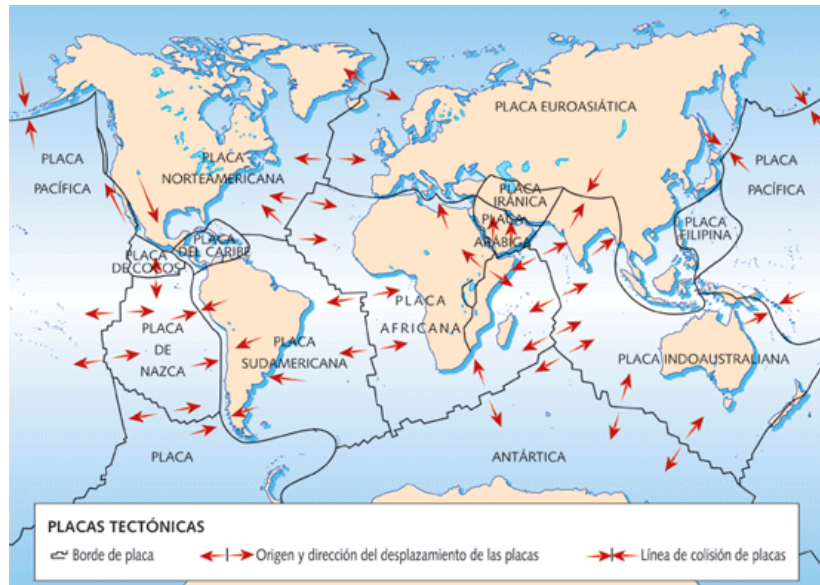


Figura 3.7: Mapa de Placas tectónicas y sus desplazamientos.

Desde el punto de vista interior donde se ha producido la fractura parte una sacudida que llega a la superficie de la tierra y origina un estremecimiento del suelo: un terremoto o sismo. Han recibido el nombre de sismos tectónicos (figura 3.8) porque están relacionados con la arquitectura del globo, porque originan el relieve terrestre.

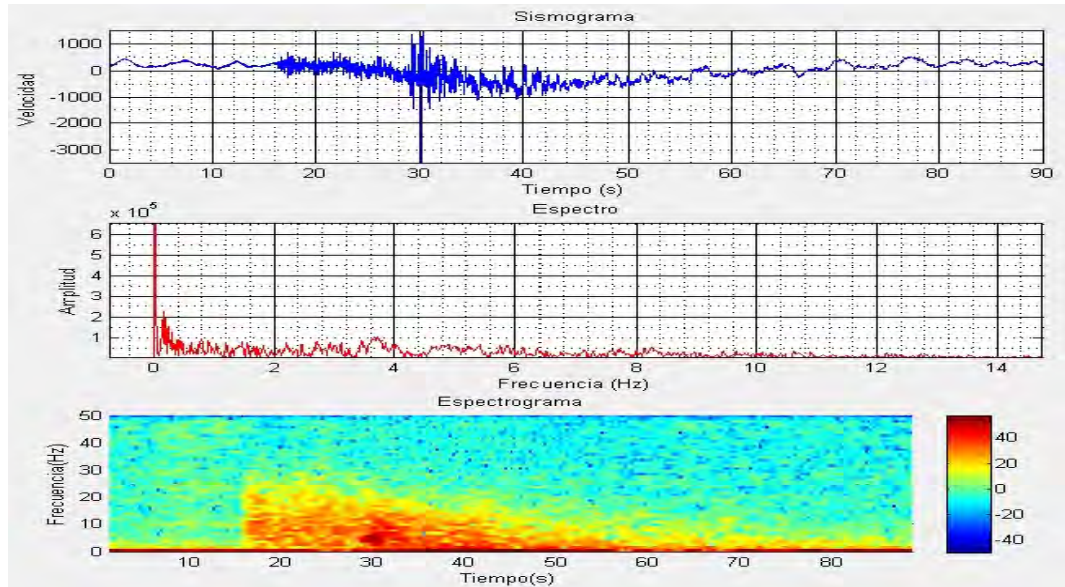


Figura 3.8: Sismograma, espectro y espectrograma de un evento tipo tectónico local en la de la estación Cobanegra.

De estos sismos, se han definido dos clases: los interplaca, ocasionados por la interacción en las zonas de contacto entre placas, y los intraplaca, que se generan en la parte interna de las placas, aun en zonas donde se ha llegado a suponer un nivel nulo de sismicidad. Estos sismos, menos frecuentes que los interplaca, pueden tener profundidades similares a éstos ($15 - 30\text{km}$) o mayores ($60 - 70\text{km}$). Cabe mencionar que los sismos de mayor magnitud que se han presentado en el mundo han sido sismos tectónicos.

3.2. Coeficientes de Predicción Lineal (LPC)

3.2.1. Modelo de Análisis por LPC

Supongamos una señal de excitación $u(n)$; esta señal está afectada por un sistema que modelaremos a través de su función de transferencia $H(z)$; la señal de salida del sistema es $s(n)$. El sistema, representado por $H(z)$, incluye características de fuente y todos los factores que afectan la señal hasta obtener el registro sismográfico (Figura 3.9).

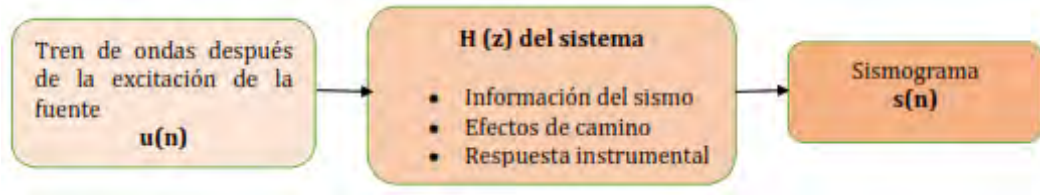


Figura 3.9: Flujo de la señal desde su excitación hasta su salida.

Los LPC se calculan utilizando un algoritmo que actúa sobre la señal $s(n)$ y arroja los vectores que contienen los coeficientes de predicción y una señal de error $e(n)$. Es posible recuperar la señal estimada $\tilde{s}(n)$ a partir de la señal de error aplicando sobre $e(n)$ un filtro inverso que contenga los LPC como coeficientes del filtro (Figura.3.10).

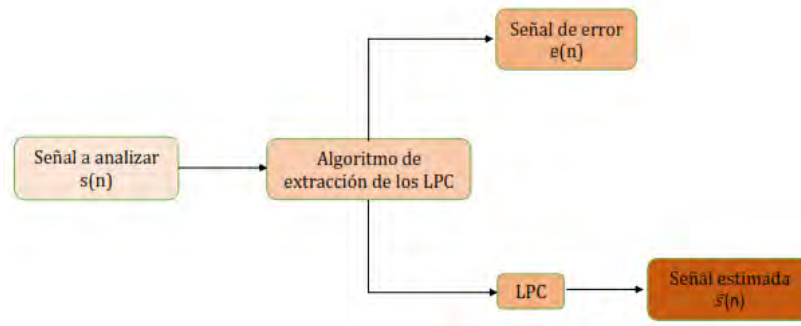


Figura 3.10: Elementos involucrados en la extracción de los LPC

La muestra $\tilde{s}(n)$ se obtiene mediante la combinación lineal de las salidas pasadas. Esta representación corresponde a la siguiente ecuación de diferencias (Ec. 3.1):

$$\tilde{s}(n) = \sum_{k=1}^p \alpha_k s(n - k) \tag{3.1}$$

El error cometido en la aproximación será:

$$e(n) = s(n) - \tilde{s}(n) \tag{3.2}$$

Donde:

- α_k : Coeficientes de predicción lineal
- $s(n)$: Señal real
- $\tilde{s}(n)$: Señal predicha
- $e(n)$: Error de predicción lineal (error residual o residuo).

El error de predicción lineal se puede escribir:

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{k=1}^p \alpha_k s(n - k) \quad (3.3)$$

Aplicando la transformada Z (TZ) a la ecuación 3.3 se tiene:

$$E(z) = TZ \left(s(n) - \sum_{k=1}^p \alpha_k s(n - k) \right) = S(z) - \sum_{k=1}^p S(z) \alpha_k z^{-k} = S(z) \left(1 - \sum_{k=1}^p \alpha_k z^{-k} \right) \quad (3.4)$$

La transformada Z de una señal de tiempo discreto $x(n)$ está definida como la serie de potencias:

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) z^{-n} \quad (3.5)$$

Donde z es una variable compleja.

se denomina $A(z)$ a la expresión:

$$A(z) = \frac{E(z)}{S(z)} = 1 - \sum_{k=1}^p \alpha_k z^{-k} \quad (3.6)$$

Es decir, $A(z)$, al que denominaremos filtro inverso, será la función de transferencia de un sistema de la forma:

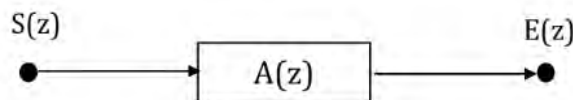


Figura 3.11: Sistema con filtro $A(z)$

Se puede observar que el filtro inverso al anterior es:

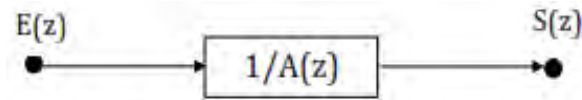


Figura 3.12: Sistema con filtro $1/A(z)$

De tal forma que si se excita con el error de predicción un sistema cuya función de transferencia sea $1/A(z)$, obtendremos a la salida la señal deseada $s(n)$.

En el modelo simplificado de producción, podemos asumir que $H(z)$ sigue un modelo todo-polos con q polos, es decir:

$$H(z) = \frac{S(z)}{U(z)} = \frac{G}{1 - \sum_{k=1}^q a_k z^{-k}} \quad (3.7)$$

Identificando $H(z)$ con el filtro de predicción lineal $\frac{1}{A(z)}$, imponiendo que el factor de ganancia G sea igual a 1 y asumiendo que el número de polos del modelo es igual al orden de predicción lineal, $p = q$, tendremos:

$$a_k = \alpha_k; H(z) = \frac{1}{A(z)}; E(z) = U(z).$$

3.3. Redes Neuronales Artificiales (RNA)

Las redes neuronales artificiales (se denotan RNA) nacen como el producto de un intento por emular ciertas características del cerebro humano desde el punto de vista sensorial. Las neuronas, procesadores elementales y relativamente lentos se agrupan formando capas y estas a su vez forman redes que funcionan como un todo que nosotros conocemos como cerebro. El estilo de procesamiento que el cerebro utiliza es en paralelo, esto es, varias neuronas pueden activarse al mismo tiempo sin que el sistema colapse por ello. El conocimiento está ampliamente distribuido en la red lo que evita grandes pérdidas de información en caso de daños parciales. El cerebro humano tiene la capacidad de adaptarse a las condiciones cambiantes del medio exterior, aprendiendo nuevas cosas y aplicándolas sobre ese nuevo entorno. En el cerebro existe una gran tolerancia a los fallos ya que si una neurona deja de funcionar no es motivo para que el sistema completo interrumpa su funcionamiento. El tipo de control de los procesos es auto-organizado, lo que implica que existen grupos de neuronas especializadas en tipos específicos de procesos.

En la descripción anterior, de las características del cerebro que se pretende emular mediante RNA se encuentran las siguientes: procesamiento en paralelo, almacenamiento distribuido, adaptabilidad, tolerancia a fallos y control de proceso auto-organizado.

Las RNA se han convertido en una alternativa para solucionar problemas con las siguientes características (Del Brío, 2002):

- Falta de un conjunto de reglas sistemáticas que describan completamente el problema.
- Disponibilidad de muchos ejemplos o casos históricos.
- Datos imprecisos o ruidosos.
- Elevada dimensionalidad.
- Condiciones de trabajo cambiantes.

3.3.1. La Neurona Artificial

Una neurona artificial es una unidad elemental de proceso que realiza cálculos a partir de un vector de entrada procedente del exterior o de otras neuronas. La neurona artificial es la base estructural de las redes neuronales artificiales[14].

En una neurona i pueden distinguirse los siguientes componentes (Figura 3.13):

- Conjunto de **entradas** x_j .
- **Pesos sinápticos** de la neurona iw_{ij} , que representan la fuerza de la interacción entre cada neurona presináptica j (anterior a los pesos sinápticos) y la neurona postsináptica i (posterior a los pesos sinápticos).
- **Regla de propagación** que proporciona el valor del potencial postsináptico de la neurona i en función de sus pesos sinápticos y entradas.
- **Función de activación** que proporciona el estado de activación actual de la neurona i en función de su estado anterior y de su potencial postsináptico.
- **Función de salida** que proporciona la salida actual de la neurona i en función de su estado de activación.

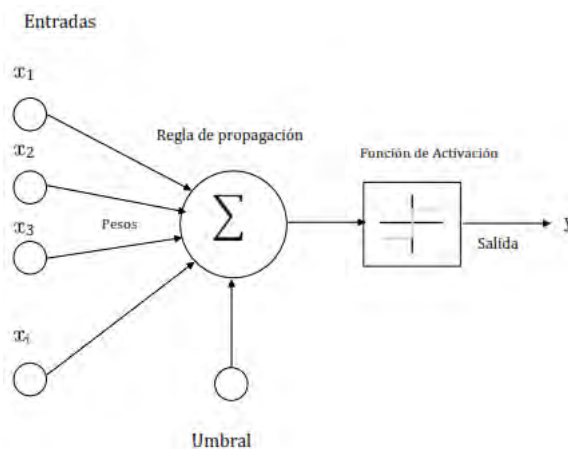


Figura 3.13: Neurona Artificial

Entradas y salidas

La forma de las entradas y salidas en las redes neuronales puede variar dependiendo del tipo de aplicación. Cuando se quiere resolver un problema de clasificación es conveniente utilizar como entradas un vector que contiene información acerca del patrón que se desea discriminar y como salidas vectores de baja dimensionalidad cuyos elementos suelen estar en los rangos $0, +1$ o $-1, +1$. Los vectores de entrada pueden también normalizarse teniendo en cuenta los conjuntos completos de los diferentes tipos de patrones a clasificar.

Pesos sinápticos

Los pesos sinápticos condicionan el poder de la interacción entre la neurona presináptica j y la neurona postsináptica i . Si la entrada es positiva el peso sináptico que interactúa con dicha entrada producirá efectos que realzarán la excitación (peso positivo) o inhibición (peso negativo) de la neurona dependiendo de si el peso es positivo o negativo respectivamente.

Regla de Propagación

La regla de propagación determina el potencial postsináptico $h(t)$ a partir del vector de entrada y los pesos sinápticos, generalmente se utiliza el producto escalar entre el vector de entrada y el de pesos sinápticos:

$$h_i(t) = \sum_j w_{ij}x_j \quad (3.8)$$

Función de Activación

De esta función depende el estado de activación de la neurona, su argumento es el potencial postsináptico. Las funciones utilizadas suelen ser continuas y monótonas crecientes; en problemas de clasificación los algoritmos utilizados para el aprendizaje de la red exigen también que la función de activación sea derivable. El estado de activación actual de la red sería entonces[14]:

$$a_i(t) = f_i(h_i(t)) \quad (3.9)$$

Función de Salida

Esta función es la que presenta el estado global de la neurona, depende del estado de activación; lo más común es utilizar este estado como la propia salida de la neurona, por tanto:

$$y_i(t) = F_i(a_i(t)) = a_i(t) \quad (3.10)$$

3.3.2. Arquitectura de las Redes Neuronales

Se denomina arquitectura de una red neuronal a la estructura que conecta las neuronas entre sí para formar una red neuronal. Las arquitecturas de red son de diversos tipos, unas son más apropiadas que otras para la solución de determinados problemas. La arquitectura de red que se escoja depende de las entradas que se usen para representar los patrones y de la manera como se desee obtener las salidas. En nuestro caso, en que se necesita clasificar entre conjuntos con diferentes características, se suele utilizar la arquitectura **perceptrón multicapa** (Del Pezzo, 2003; Rugeles Cadena 2006;Dimaté Cadena 2011).

En la arquitectura perceptrón multicapa las neuronas se agrupan en unidades denominadas capas, una capa generalmente contiene neuronas de un mismo tipo. El número de capas y neuronas por capa se denomina topología de la red. Las capas son de tres tipos: de entrada, de salida y ocultas. Las capas de entrada y salida tienen contacto directo con el exterior, es decir, se comportan como sensores, en el caso de las capas de entrada, o afectores en el caso de las capas de salida. Las capas ocultas no tienen contacto con el exterior, pero se comportan como los elementos computacionales más importantes al momento de buscar las relaciones entre las entradas y las salidas deseadas[14].

Una vez conectadas las capas para formar una red neuronal es posible que la información fluya únicamente hacia adelante, es decir, desde las neuronas de entrada hacia las ocultas y

finalmente a las de salida (Figura. 3.14); Para este trabajo se usan este tipo de redes llamadas unidireccionales, pero también puede suceder que la información fluya de tal manera que se devuelva entre las capas o inclusive desde la capa de salida hacia la de entrada, en este caso estaremos tratando redes neuronales recurrentes.

3.3.3. Definición de Red Neuronal Artificial

Una definición formal de red neuronal es utilizando el concepto de grafo (Muller, 1990) así:

Un grafo G es un par ordenado $G = (V, E)$, donde:

V es un conjunto de vértices o nodos. Normalmente V suele ser finito.

E es un conjunto de aristas o arcos, que relacionan estos nodos.

Un grafo dirigido es un grafo $G = (V, E)$ donde:

$$V \neq \emptyset$$

$E \subseteq ((a, b) \in V \times V : a \neq b)$ es un conjunto de pares ordenados de elementos de V .

Dada una arista (a, b) , a es su nodo inicial y b su nodo final. Por definición, los grafos dirigidos no contienen bucles.

Una red neuronal es un grafo dirigido con las siguientes propiedades:

- A cada nodo i se asocia una variable de estado x_i .
- A cada conexión (i, j) de los nodos i y j se asocia un peso $w_{ij} \in R$.
- A cada nodo i se asocia un umbral Θ_i .

- Para cada nodo i se define una función $f_i(x_i, w_{ij}, \theta_i)$ que depende de los pesos de sus conexiones, del umbral y de los estados de los nodos j a él conectados. Esta función proporciona el nuevo estado del nodo.

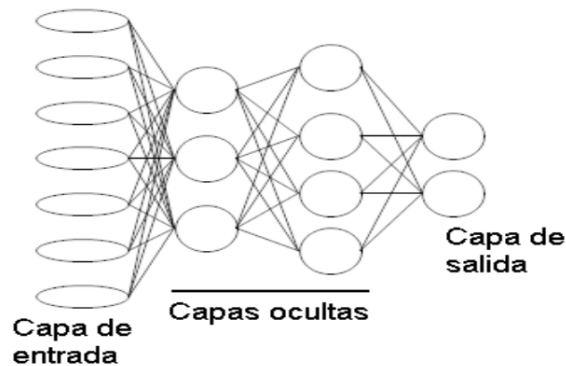


Figura 3.14: Red Neuronal Artificial

3.3.4. Operación de las Redes Neuronales Artificiales

Existen dos modos en los que operan las RNA, *aprendizaje o entrenamiento* y *recuerdo o ejecución*. En la fase de entrenamiento la RNA aprenderá una tarea deseada hasta que el entrenador (quien opera la red) considere que la tarea puede ser ejecutada sin la necesidad de ser supervisada, es entonces cuando se pone en marcha la fase de ejecución en que la RNA entrenada aplica los conocimientos adquiridos para realizar la labor para la que fue creada[13].

Fase de Aprendizaje

A nivel operativo la fase de aprendizaje o entrenamiento consiste en ajustar los pesos sinápticos w_{ij} de la red a través de la aplicación de un algoritmo que minimiza una función de error hasta satisfacer un criterio establecido anticipadamente. Este algoritmo llamado también *regla de aprendizaje* es el encargado de tomar las señales del exterior o ejemplos

y en función de estos ajustar en un valor Δw_{ij} el estado actual de los pesos sinápticos:

$$w_{ij}(t + 1) = w_{ij}(t) + \Delta w_{ij}(t) \quad (3.11)$$

Donde $w_{ij}(t)$ representa el valor actual del peso sináptico que une la neurona presináptica j con la postsináptica i y Δw_{ij} representa el ajuste. Este proceso se realiza de manera iterativa hasta que se considere que la red ha alcanzado cierto nivel de aprendizaje. En este punto del proceso la RNA tiene una percepción del entorno que utilizará para realizar la tarea que le fue asignada. De los tipos de aprendizaje que existen los más utilizados son el **supervisado** y el no supervisado.

Aprendizaje Supervisado

En este tipo de aprendizaje el entrenador le presenta a la red patrones de un tipo junto con una etiqueta del conjunto al que pertenece el patrón. Por ejemplo, se presenta un conjunto de vectores como entrada y un objetivo común como salida; este último suele ser un vector de menor dimensión con el cual se desea asociar las entradas. El proceso de presentación de los patrones es iterativo y la regla de aprendizaje ajusta los pesos sinápticos de la red en cada iteración hasta encontrar una relación entre las entradas y las salidas.

Retropropagación del Error

Dependiendo del tipo de problema a resolver existe una variedad de algoritmos de aprendizaje, entre los más utilizados están: el aprendizaje Hebbiano, la regla LMS (Least Mean Squares), la regla de aprendizaje de Boltzman, el de corrección de errores que incluye la Adalina y el de retropropagación del error[14].

Sea E una función de error $E[W]$, $E = R^n \rightarrow R$, función de los pesos sinápticos que mide la eficacia de la red. Se requiere encontrar un proceso que optimice la función de error buscando un mínimo. Para ello se utiliza el método de **descenso por el gradiente** (Figura. 3.15) Si la función utilizada dependiera solamente de dos variables su gráfica sería semejante a una

sábana plegada con crestas y valles; estar sobre uno de sus valles corresponde a estar ubicado en un mínimo local óptimo. El objetivo, que no siempre es posible de lograr, es encontrar un mínimo global; sin embargo un buen mínimo local muy frecuentemente es suficiente para abordar el problema. Este proceso se lleva a cabo iterativamente con ayuda de los patrones de aprendizaje y los objetivos de cada conjunto de entrenamiento.

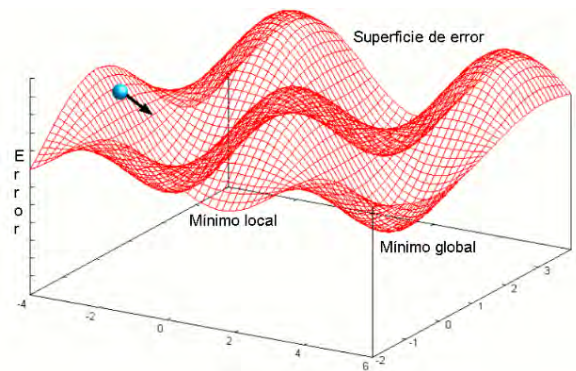


Figura 3.15: Esquema de descenso por el gradiente para una función de error. La esfera azul indica el punto donde se evalúa el error y la flecha la dirección hacia donde se debe mover la esfera para alcanzar un mínimo de la función (Tomado de JOONE, 2007).

En el método de descenso por el gradiente los pesos sinápticos para $t = 0$, $W(0)$ (inicializados con valores aleatorios pequeños) se ajustan una vez se ha calculado el gradiente de la función de error $E(W)$. El gradiente proporciona la dirección de la máxima variación de la función, por tanto el ajuste se hace en sentido contrario a esta dirección y en un valor adecuado, no tan grande de tal manera que no se salten características importantes de esa superficie tales como mínimos locales cercanos, ni tan pequeño que el tiempo de computo sea excesivamente largo. Cada una de estas actualizaciones tendría la forma:

$$W(t + 1) = W(t) - \epsilon \nabla E(w) \quad (3.12)$$

donde ϵ indica el tamaño del paso tomado en cada iteración. Para la iteración t la variación en la función de error $E(W)$ es:

$$\delta(E(w_{ij})) = \sum_{ij} \frac{\delta E(w_{ij})}{\delta w_{ij}} \delta w_{ij} \quad (3.13)$$

donde los w_{ij} son los elementos de la matriz de pesos sinápticos. Al multiplicar por el gradiente obtenemos:

$$\delta(f(w_{ij})) = \sum_{ij} \left(\frac{\delta E(w_{ij})}{\delta w_{ij}} \right) \left(-\epsilon \frac{\delta E(w_{ij})}{\delta w_{ij}} \right) \quad (3.14)$$

Así se demuestra que la actualización es siempre menor que cero, asegurando que a través de este método se encontrarán mínimos ya sean locales o globales.

En este método la forma como se alcanza un mínimo es asintótica, ya que con el transcurrir de las iteraciones las variaciones son cada vez más pequeñas dado que son proporcionales al error asociado al paso por la red.

La desventaja de la regla LMS reside en que la función de error debe estar en función únicamente de los pesos sinápticos de la capa de salida. En el caso de una arquitectura neuronal con capas ocultas es necesario generalizar la regla, esto se logra con el algoritmo BP. Debido a que en el proceso de ajuste de los pesos sinápticos es necesario derivar las funciones según la regla de la cadena se exige que las funciones de transferencia de las neuronas sean derivables.

Aceleración del aprendizaje por retropropagación de Error (BP)

Para resolver algunos de los inconvenientes del BP se plantean continuamente correcciones o variantes. Buena parte de esas modificaciones tratan de resolver el problema de la lenta convergencia. Así la primera variante la propusieron los propios inventores del método (Del Brio, 2002), al incluir en el algoritmo un término denominado **momento**, consistente en añadir al cálculo de la variación de los pesos un término adicional proporcional al incremento de la iteración anterior (ecuación 3.15).

$$\delta w_{ij}(t+1) = -\epsilon \frac{\delta E(w_{ij})}{\delta w_{ij}}(t) + \alpha \cdot \delta w_{ij}(t-1) \quad (3.15)$$

Con α un parámetro entre 0 y 1, que suele tomar próximo a 1 ($\alpha = 0,9$) (Del Brío, 2002). De esta manera, si los incrementos en un determinado peso tienen el mismo signo, las actualizaciones en cada iteración serán mayores: sin embargo, si los incrementos en cierto peso oscilan (a veces positivos, otras negativos), el incremento acumulado se reduce o cancelarse. Así, en zonas estrechas y profundas de la hipersuperficie del error, los pesos correspondientes a las dimensiones estrechas (que sin el término de momento oscilarían de un lado al otro del valle) sufren incrementos pequeños, mientras que los de las direcciones que descienden directamente al fondo se ven potenciados (Bishop, 1994). Es esta una manera de incrementar el ritmo de aprendizaje efectivo en determinadas direcciones [14].

Fase de Recuerdo

Una vez ha concluido la fase de aprendizaje, la RNA está lista para responder a estímulos con los que nunca ha trabajado, pero que tienen características comunes a los conjuntos con los que se entrenó. Inicia entonces la fase de recuerdo, que depende de la manera en que están memorizadas las características de los patrones con que se entrenó a la red.

3.3.5. Capacidad de Generalización

Si el problema a solucionar fuese el de verificar que el patrón desconocido es exactamente uno de los presentados en la fase de entrenamiento, sería mejor utilizar una base de datos que contuviera todos los vectores con los que se habría aplicado el entrenamiento y un algoritmo que comparara elemento a elemento los vectores memorizados con el que se desea dar respuesta. Una de las ventajas de las RNA es que no necesita memorizar uno a uno los patrones de entrenamiento sino el de reconocer características claves en ellos y extender ese conocimiento a estímulos desconocidos, esto es lo que se conoce como generalización. Para

detener el entrenamiento es necesario medir el nivel de *generalización* de la red en función de su nivel de aprendizaje[13].

3.3.6. Validación cruzada

Durante el proceso de entrenamiento aparecen dos tipos de error:

- el error de ajuste de cada patrón a su respectivo objetivo, llamado error global, que se calcula como el error cuadrático medio del conjunto de patrones y objetivos (para el caso del aprendizaje supervisado).
- el error de respuesta a patrones desconocidos, llamado *error de generalización*.

Una RNA ha tenido una fase de aprendizaje exitosa cuando extiende lo aprendido a patrones desconocidos, esto se puede evaluar realizando pruebas de la red a ejemplos de entrada no conocidos. Es preferible seleccionar una RNA que muestre un error de generalización aceptable a una que arroje errores globales pequeños, ya que estos últimos pueden reducirse tanto como se quiera con solo incrementar el número de iteraciones. Un hecho observado en experimentos con simuladores de RNA es que cuando se alcanza un error global bastante pequeño, digamos menor que 1

El error global disminuye asintóticamente a medida que el número de iteraciones crece, mientras que el error de generalización alcanza un mínimo. Detener el proceso de entrenamiento o guardar la configuración de pesos sinápticos y umbrales en ese instante en que el aprendizaje es óptimo, nos asegura que la configuración de la matriz de memoria es la más adecuada para utilizarla en la fase de recuerdo, esta técnica se conoce como validación cruzada (Figura 3.16).

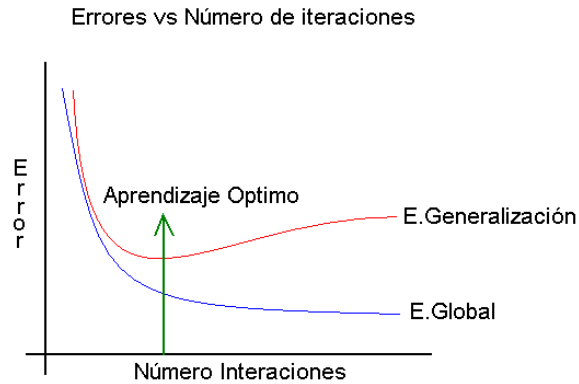


Figura 3.16: Errores Global y de generalización en un problema ideal, la flecha verde indica el momento en que el aprendizaje es óptimo (Tomado de CADENA, 2011)[14].

3.3.7. Perceptrón Multicapa

El perceptrón multicapa o MLP (Multilayer perceptron) tiene sus raíces en una arquitectura de red más simple denominada perceptrón simple, que consta de dos capas, una de entrada, que no realiza cálculo alguno y otra de salida que es la que realiza las operaciones encargadas de dar respuesta a un estímulo determinado. La función de activación de un perceptrón simple es la función escalón o Heaviside (Figura 3.17):

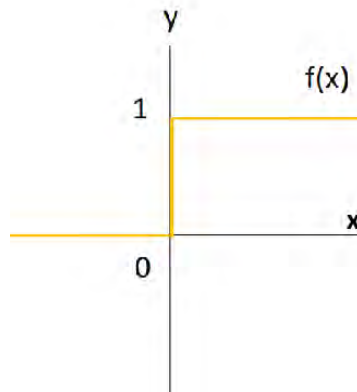


Figura 3.17: Representación gráfica de la función escalón o Heaviside.

$$f(x) = \begin{cases} 1, & \text{si } x > 0 \\ 0, & \text{si } x < 0 \end{cases}$$

(3.16)

Esta arquitectura básica puede utilizarse para problemas simples de clasificación debido a que su salida es esencialmente booleana. La gran limitación del perceptrón simple es que puede discriminar dos conjuntos *linealmente separables*, esto es, que se puedan distinguir con ayuda de un hiperplano para el caso de más de dos dimensiones o una línea recta para el caso de dos dimensiones.

El MLP supera las limitaciones del perceptrón simple y permite abordar problemas complejos de clasificación para conjuntos no necesariamente linealmente separables. Esta RNA se entrena con ayuda del algoritmo de retropropagación del error y tiene la característica de que la función de activación que se utiliza en sus neuronas ocultas y de salida es de tipo sigmoideo (Figura. 3.18)[14].

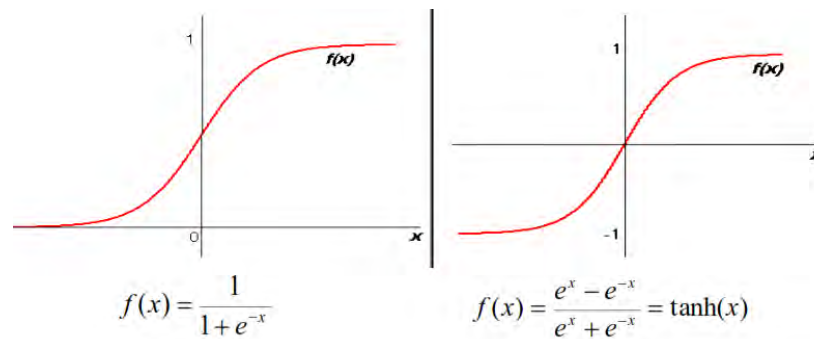


Figura 3.18: : Funciones de activación utilizadas en las neuronas del MLP. (a) a la izquierda la correspondiente a la función log-sigmoidea y a la derecha la función tangente-sigmoidea (Tomado de CADENA, 2011).

El entrenamiento de un MLP, siguiendo el procedimiento que aplica el algoritmo de retropropagación del error, se resume en los siguientes pasos:

- Tras inicializar los pesos de forma aleatoria y con valores pequeños, se selecciona el primer par de entrenamiento (Entradas - objetivos).
- Se calcula la salida de la red
- Se calcula la diferencia entre la salida real de la red y la salida deseada, con lo que se obtiene el vector de error.
- Se ajusta los pesos de la red de forma que se minimice el error.
- Se repite los pasos anteriores para cada par de entrenamiento hasta que el error para todos los conjuntos de entrenamiento sea aceptable.

Existen variaciones al algoritmo en las que se reemplaza el método de descenso por el gradiente por otras técnicas llamadas de segundo orden como son los gradientes conjugados, gradientes conjugados escalados y **Levenberg Marquardt**, que agilizan la convergencia hacia un mínimo. El objetivo principal del uso de estas variantes es hacer más eficiente el recurso computacional, pero en términos de la efectividad alcanzada, en general, ninguno es superior (Del Brío, 2002).

3.3.8. Numero de ejemplos de entrenamiento

(Haykin, 1999) afirma que la capacidad de generalización de la red la determinan en buena medida las siguientes tres circunstancias:

- Arquitectura de la red.
- Número de ejemplos de entrenamiento.
- Complejidad del problema.

(Baum, 1989) demostró que para una red con n entradas y h neuronas ocultas con w pesos en total, se requiere un número de patrones de aprendizaje del $p = w/\epsilon$ para proporcionar un error de aprendizaje del orden de ϵ . De acuerdo a lo anterior, de tal manera que, para un MLP

con 15 neuronas en la capa de entrada, cinco en la capa oculta y dos en la de salida (15-5-2) el número de ejemplos necesarios para entrenar la red será de 1840[2].

(Haykin, 1999) sugiere reservar un 20 % de los datos para aplicar la técnica de validación cruzada. Para un número de neuronas de entrada elevado, el número de parámetros de la red lo será aún mucho más, de tal forma que se necesitará un número de patrones de aprendizaje que en la mayoría de los casos es imposible de tener. Por ejemplo: si se cuenta con una red de 200 neuronas de entrada, cinco en su capa oculta y 2 en la salida (200 – 5 – 2) el número de ejemplos de entrenamiento para lograr un adecuado aprendizaje (digamos con un error del 5

La solución de este problema consiste en realizar un pre-procesamiento de los patrones de entrenamiento y así reducir la dimensión del vector de entrada; para ello se utilizan técnicas que reduzcan y representen de manera adecuada la señal original.

Capítulo 4

Metodología

Para la realización de esta investigación se requirió de cinco pasos: En el primer paso se plantea cómo abordar la discriminación de los sismos, los tipos de sismos a clasificar y las señales que están fuera del alcance de este trabajo; en el segundo se representa y reduce la dimensión de las señales de entrada, y para esto se estudia el orden de la predicción lineal para así tener una adecuada representación de las señales; en el tercero se diseña y desarrolla un programa el cual facilite la manipulación de las señales para su posterior discriminación y se implementa una base de datos relacional para organizar la información generada por el programa; en el cuarto paso se aborda el estudio de las topologías de las RNA para la clasificación, y esto se logra mediante el entrenamiento de las RNA con diferentes topologías las cuales posteriormente se ejecutan para evaluar los resultados; en el quinto paso se procede a la ejecución del proceso de clasificación. Como ejemplo de prueba de la metodología y de la herramienta computacional se realiza un ejercicio de clasificación de eventos en una de las estaciones de la red sismológica del volcán galeras pero su aplicación se puede extender a todas las estaciones de la red. Toda la parte computacional de este trabajo se realiza en el programa MATLAB.

4.1. Discriminación de sismicidad

Las RNA funcionan como clasificadoras, de tal forma que se va a clasificar en dos grupos de señales, las que están asociadas a sismicidad tectónica (sismos tectónicos locales) y las que están asociadas a los sismos volcánicos (sismos tipo LP, VT-A, VT-B, TRE). Los elementos de cada conjunto corresponden a segmentos de traza, que contienen un número determinado de muestras. La duración de cada segmento se mide en segundos y ésta será un parámetro

manipulable en el programa.

Pre-procesamiento de las señales

La cantidad de muestras utilizadas son 100 por segundo (Figura 4.1) aunque se sabe que los sismogramas de banda ancha no contienen en su mayoría frecuencias superiores a los 35 Hz.

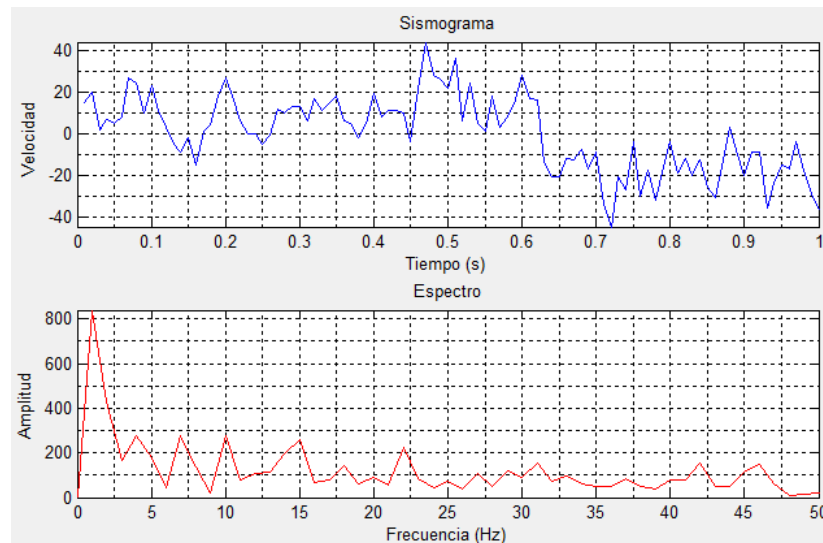


Figura 4.1: Sismograma y espectro muestreado a 100 Hz.

En este pre-proceso se cortan las señales en segmentos del mismo número de muestras, se remueve el offset, se aplica un filtro pasa-banda entre 5 y 30 Hz y se normalizan las unidades, esta normalización se hace en el rango $[-1,1]$.

Por ultimo se aplica sobre los segmentos de traza un proceso de ventaneo o Hamming (Figura 4.3). Se sabe que la señal correspondiente a un sismo es no estacionaria, sin embargo, como el análisis de la señal se realiza en ventanas cortas de tiempo se puede suponer que los tramos son estacionarios dado que el contenido espectral de la señal varía lentamente en un lapso corto de tiempo. Se emplea una ventana suavizada en sus extremos como la de Hamming para evitar el efecto de borde (Cadena, 2011):

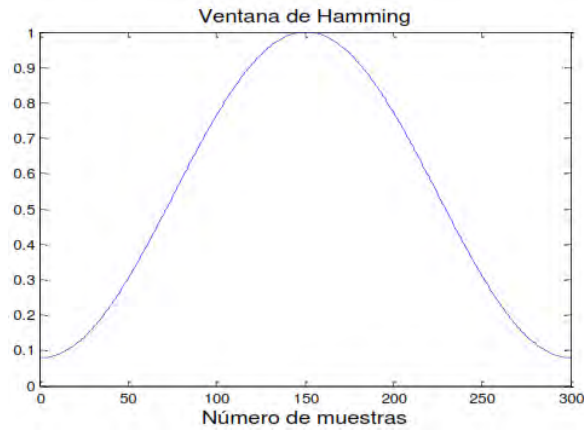


Figura 4.2: Ventana Hamming para 300 muestras.

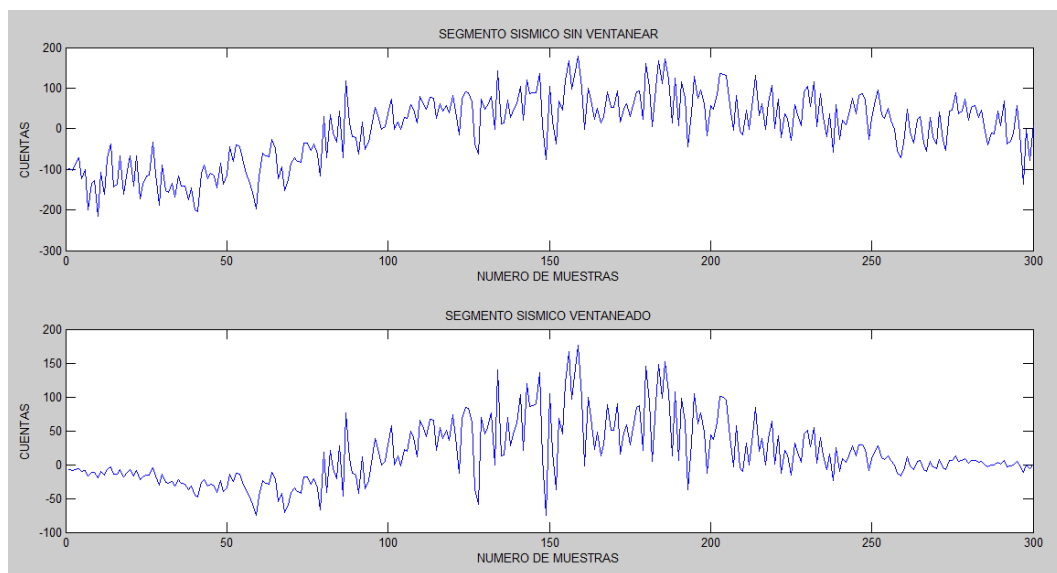


Figura 4.3: (Arriba) se muestra un tramo de evento sísmico sin ventanear y (abajo) el mismo segmento ventaneado con la función de Hamming.

4.2. LPC para la representación de las señales

Aplicar la técnica LPC es una parte importante de este trabajo porque con esta se reduce la dimensión de los vectores de entrada al sistema discriminador.

La escogencia del orden de la predicción lineal debe ser tal, que el número de coeficientes no sea tan grande como para incluir detalles de la señal que entorpecerían la discriminación ni tan pequeño que resulte en la pérdida de información importante de la señal.

Tamaño de los segmentos de traza

Para observar una frecuencia mínima de 0.5 Hz se debe tener una ventana de 2 segundos de registro, sin embargo, los espectros de señales capturadas con sensores banda ancha muestran frecuencias menores que ese valor. Si ponemos una ventana de 3 segundos de duración nos permitiría ver armónicos con frecuencias de hasta 0.33 Hz, siendo este un valor aceptable para el espectro de los registros sísmicos utilizados en este trabajo ya que a todos los datos se les aplica un filtro pasa-banda que tiene como valor mínimo 5 Hz.

En la herramienta computacional es posible elegir el tiempo de corte, pero se debe tener en cuenta que para tamaños más grandes de ventana mejorarían la resolución en el dominio de la frecuencia, pero incrementarían el esfuerzo computacional del sistema, volviéndolo lento.

4.3. Creación del algoritmo en MATLAB e implementación del programa *DSTYV*

El programa *DSTYV* (*Discriminador de Sismicidad Tectónica Y Volcánica*) fue desarrollado con la participación del M.Sc. Oscar Cadena Ibarra (OVSP) y el estudiante Jhon Jairo Ramírez. *DSTYV* está creado en MATLAB. La estructura de *DSTYV* se muestra en la Figura 4.4.

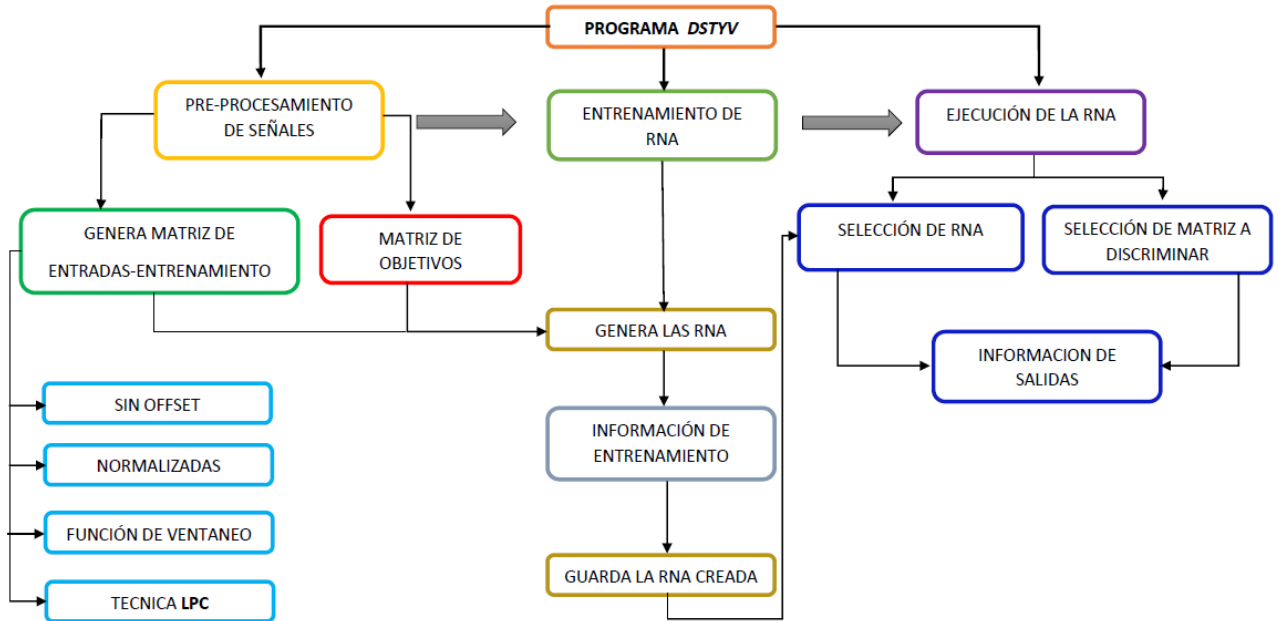


Figura 4.4: Estructura del programa *DSTYV*

El entorno del programa consta de tres módulos los cuales desempeñan diferentes funciones que a continuación se describirán de manera detallada.

Pre-procesamiento de señales

En este módulo se preparan tanto las matrices de entrenamiento como las de ejecución. En la primera fase de este módulo se seleccionan los archivos en código ASCII (American Standard Code for Information Interchange)(Figura 4.5) a pre-procesar (Figura 4.6)

```

07131201.XTL
CHAN
2015/07/13 12:01:54.8800
100.0000 m/s
13536 muestras
-31008
-31163
-31248
-31247
-31204
-31172
-31084
-30997
-30875
-30784
-30848
-30946
-31021
-31067
-31189
-31256
-31283
-31299
-31183
-31040
-31012
-30941
-30828
-30844
-30844
-30880
-30950
-31051
-31143
-31236
    
```

Figura 4.5: Formato de los archivos ASCII que contienen las trazas utilizadas

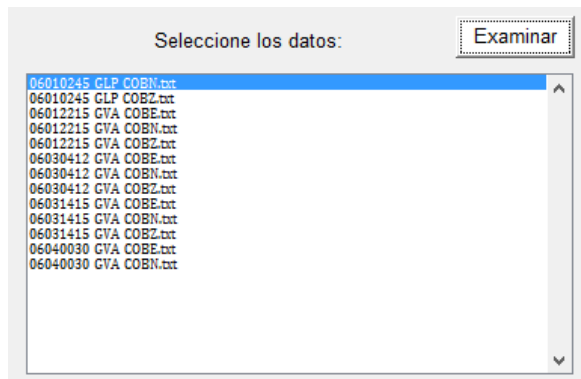


Figura 4.6: Sección de selección de trazas

En la siguiente sección (Figura 4.7) el usuario puede seleccionar el tiempo de corte de las trazas seleccionadas. Se aclara que debe ser un múltiplo de 3 ya que los segmentos de traza a aplicarles el algoritmo LPC son de 300 muestras.

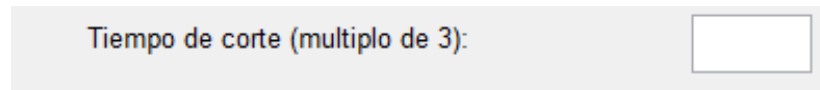


Figura 4.7: Sección de selección de tiempo de corte de las trazas

Después de elegir el tiempo de corte, las señales están listas para ser transformadas en la matriz pre-procesada. El modulo de pre-procesamiento genera dos tipos de matrices:

Una matriz (Figura 4.8) con las siguientes características:

- Sin encabezado
- sin offset
- con filtro pasa-banda.

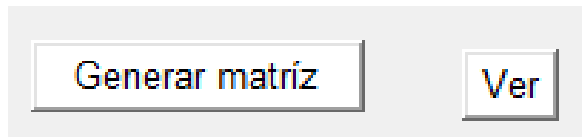


Figura 4.8: Sección de generación de matriz sin LPC

La sección *generar matriz de LPC's* (Figura 4.9), genera la matriz a utilizar en el modulo de entrenamiento y/o ejecución. En esta sección, se selecciona el orden del polinomio de los LPC. La matriz generada presenta las siguientes modificaciones:

- Sin encabezado
- Sin offset
- Con filtro pasa-banda.
- Se normalizan las columnas de la matriz.
- Se aplica ventaneo (hamming)

- A cada intervalo de 300 muestras de cada columna se le aplica la tecnica LPC.

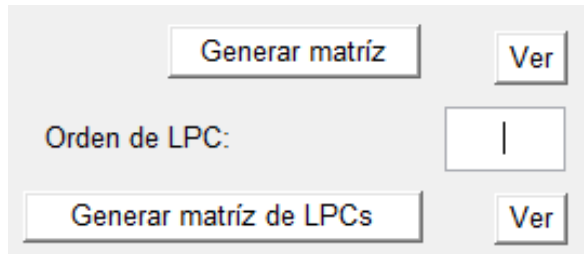


Figura 4.9: Sección de generación de matriz de LPC

De esta forma se obtiene la matriz de coeficientes de predicción lineal.

Entrenamiento de Red

En este modulo se determinan los parámetros mas importantes y necesarios para entrenar las redes neuronales.

En la primera sección, se introduce la matriz de entrenamiento , es decir, la matriz pre-procesada (Figura 4.10).

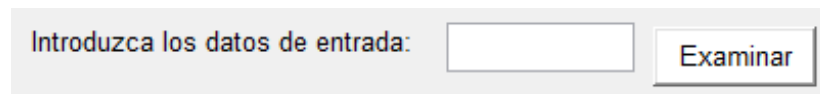


Figura 4.10: Sección de selección de matriz de entrada para entrenamiento

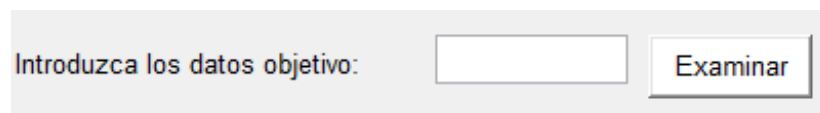
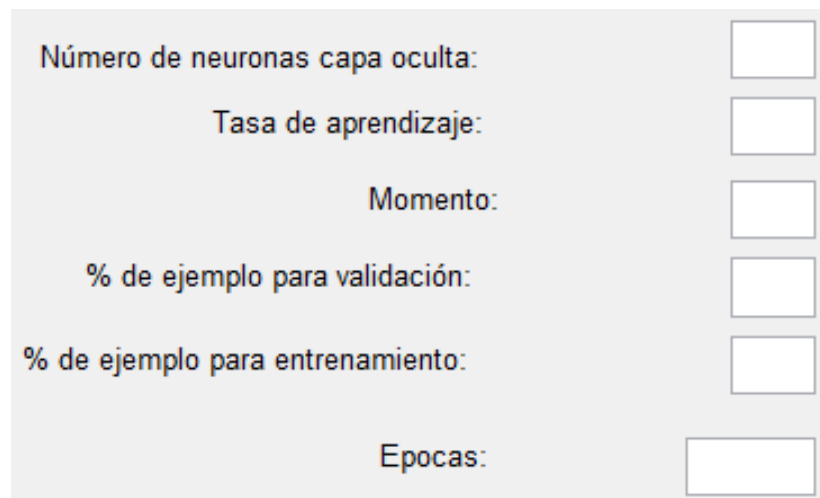


Figura 4.11: Sección de selección de matriz de objetivos para entrenamiento

Como siguiente paso, se tiene la sección de ingreso de la matriz de objetivos (Figura 4.11). Con estas dos secciones completadas, se tiene el par de entrenamiento.

La sección de parámetros de entrenamiento red 4.12, permite al usuario, la manipulación de los parámetros considerados importantes a la hora de personalizar el entrenamiento de una RNA, los cuales son:

- El número de neuronas de la capa oculta.
- La tasa de aprendizaje ϵ .
- Selección de *momento*.
- Porcentaje de los datos introducidos para tomarlos como ejemplos de validación.
- Porcentaje de los datos introducidos para tomarlos como datos de entrenamiento.
- El numero de épocas (*iteración + prueba*)



Formulario de parámetros de entrenamiento de red con los siguientes campos:

- Número de neuronas capa oculta:
- Tasa de aprendizaje:
- Momento:
- % de ejemplo para validación:
- % de ejemplo para entrenamiento:
- Epocas:

Figura 4.12: Sección de parámetros de entrenamiento de red

En la figura 4.13 se indica la sección en la cual se aplican los parámetros y se entrena la red. También hay una opción la cual permite ver la estructura de la red creada (Figura 4.14).

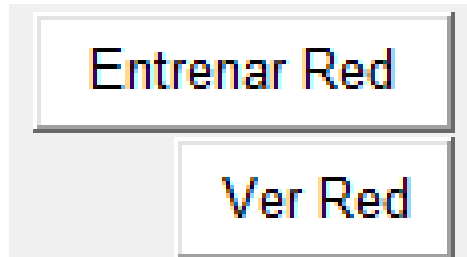


Figura 4.13: Sección de entrenamiento de red

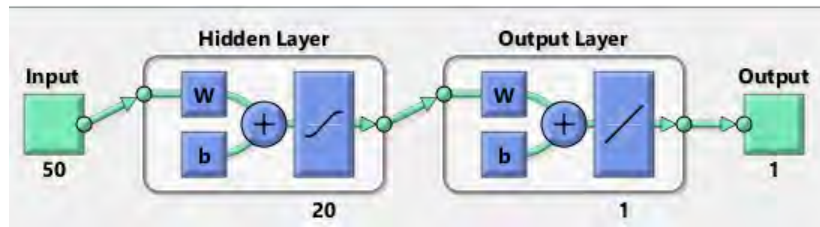


Figura 4.14: Estructura de red generada mediante *DSTYV*

En la fase de entrenamiento, el programa *DSTYV* escoge automáticamente la época en la que el error es mínimo (Figura 4.15).

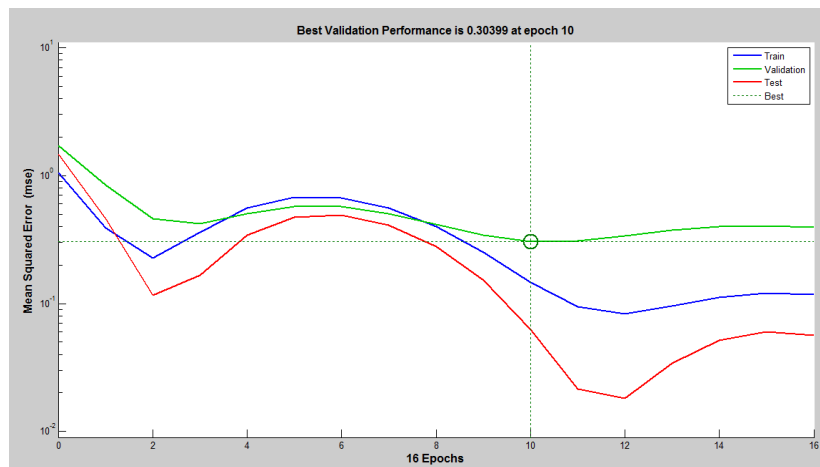


Figura 4.15: Gráfica de los diferentes errores de entrenamiento, indicando que la mejor validación ocurrió en la época numero 10.

Después del entrenamiento de la red, la sección de gráficos de estado permiten ver si el entrenamiento fue el adecuado para que la red realice lo que deseamos o si necesita un re-entrenamiento o un cambio en los parámetros de entrenamiento (Figura 4.16).

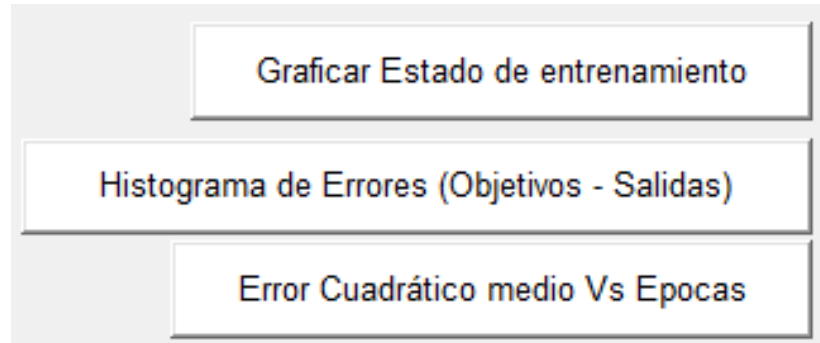


Figura 4.16: Sección de gráficos de estado de entrenamiento

Cuando se termine el proceso de entrenamiento de la RNA, la sección de almacenamiento de red (Figura 4.17) permite guardar la red generada.



Figura 4.17: Sección de almacenamiento de red

Modulo de ejecución

El modulo de ejecución tiene como finalidad la discriminación de sismos que no hicieron parte del entrenamiento, incluso, de datos de estaciones diferentes a la que se utilizó en el entrenamiento.

La primera sección (Figura 4.18) consiste en seleccionar la red de trabajo y los datos a discriminar.

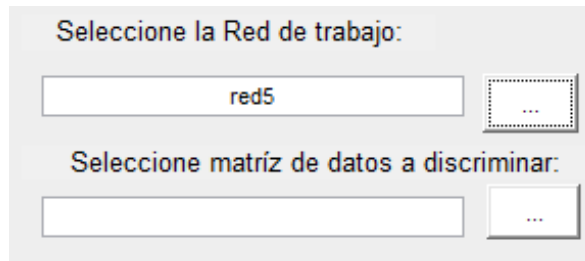


Figura 4.18: Sección de ejecución de RNA

Después de la ejecución de la RNA seleccionada por el usuario, es posible ver las salidas (Figura 4.19).

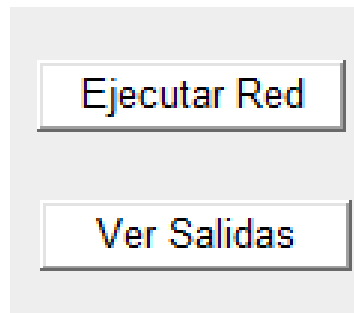


Figura 4.19: Botón para ver salidas de la RNA.

DSTYV permite también ver la estadística de las salidas obtenidas por medio de una sección de gráficas (Figura 4.20) como:

- El tipo de sismo Vs cada dato introducido
- Un histograma que indica hacia que valor tuvo tendencia cada tipo de sismo.
- Porcentaje por tipo sismo.

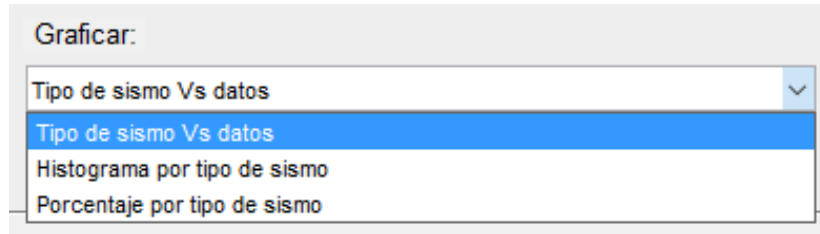


Figura 4.20: Sección de estadísticas de las salidas de RNA.

El programa *DSTYV* posee una barra de estado (Figura 4.21) que proporciona no solo la información introducida por el usuario, sino valores de parámetros predeterminados del programa y los cuales es importante mostrar.

INFORMACION DE PREPROCESO	
SIN OFFSET:	Sí
No. de muestras por toma de LPCs:	300
VENTANEADO:	Sí
TIPO DE FILTRO APLICADO:	Pasabanda
RANGO DE FILTRO:	5 - 30 Hz
DATOS NORMALIZADOS:	Sí
INFORMACION DE PROCESO	
Informacion General	
Regla de aprendizaje:	Retropropagación de error
Tipo de entrenamiento:	descenso de gradiente con momento
Entradas:	Preprocesadas
Nº de datos generados:	<input type="text" value="---"/>
Nº de datos para la validación:	<input type="text" value="---"/>
Datos de Red	
Nº de neuronas de entrada:	<input type="text" value="---"/>
Nº de neuronas capa oculta:	<input type="text" value="---"/>
Nº de neuronas capa de salida:	<input type="text" value="1"/>
Inf. de entrenamiento	
Tasa de aprendizaje:	<input type="text" value="---"/>
Momento:	<input type="text" value="---"/>
Epocas:	<input type="text" value="---"/>
Inf. de Salida	
Nombre de red generada:	<input type="text" value="---"/>
<input type="button" value="LIMPIAR"/>	

Figura 4.21: Barra de estado *DSTYV*.

4.4. Selección de topologías para la adecuada discriminación

En los trabajos de (e.g., del Pezzo, Rugeles Cadena, Cadena Arcos, Cadena Dimaté) se toma la arquitectura de tipo perceptrón multicapa con una capa oculta porque es adecuada para clasificar señales sísmicas, y los errores en sus mediciones no sobrepasaron el 10 %.

El número óptimo de neuronas en la capa oculta se determina por ensayo y error ya que no existe una regla que permita agilizar este proceso (Del Brío, 2002). Para llegar a un resultado satisfactorio se debe jugar cambiando los parámetros de entrenamiento, pero partiendo de unos valores teóricos razonables hasta llegar al valor deseado.

Capítulo 5

Resultados

5.1. Representación de las señales

5.1.1. Escogencia del orden de los LPC

El criterio para escoger los ordenes LPC adecuados para representar las señales se hace utilizando un pequeño grupo de muestras preliminares de VT, LP, TRE y de sismos tectónicos locales con los cuales se generaron diferentes matrices a distintos ordenes LPC y de esta manera se escogen aquellas que den mejores resultados respecto a sus errores global y de generalización en el proceso de entrenamiento. De esta selección se observó que los ordenes 5, 7 y 9 son los que generan los mejores resultados en el entrenamiento.

El rango inicial de los ordenes LPC a escoger fueron desde orden 3 hasta 10; Este intervalo fue tomado teniendo en cuenta estudios anteriores aplicados a datos similares a los utilizados en este trabajo (Cadena 2011).

Ademas de tener en cuenta que los ordenes escogidos deben ser los adecuados tales que no sean tan grandes para que tomen detalles sin importancia de la señal ni tan pequeños que omitan partes importantes de la misma. En la Figura 5.1 se muestra los resultados obtenidos para escoger los mejores ordenes de LPC.

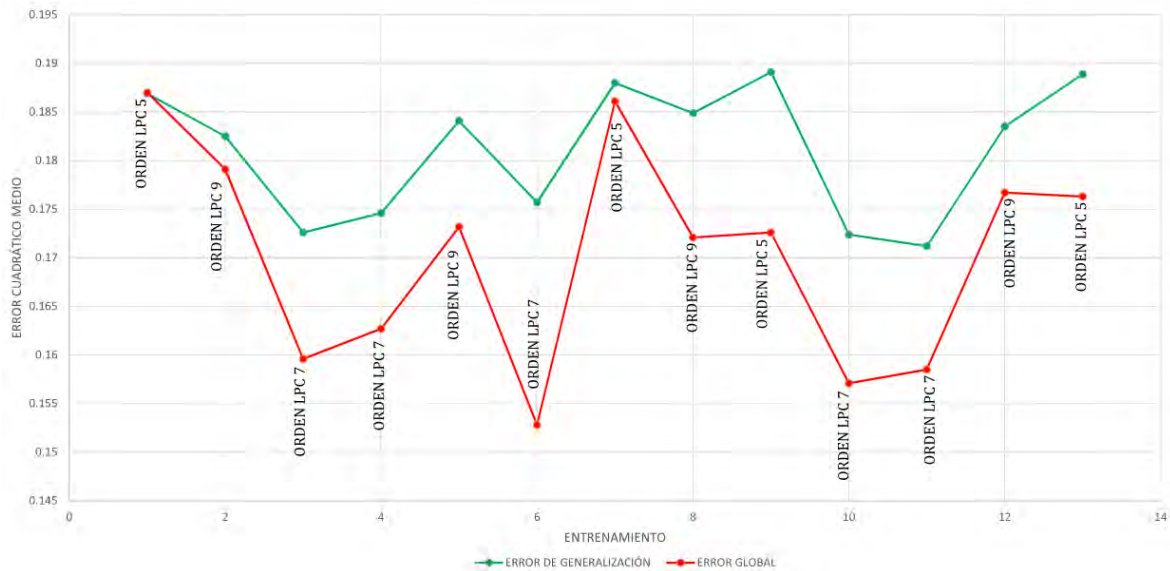


Figura 5.1: En La gráfica se muestran los tres ordenes LPC seleccionados en este trabajo. se puede observar que de los tres ordenes, el que mejores resultados proporciona es el de orden 7.

5.1.2. Duración de los segmentos de traza

La elección de las duraciones de los segmentos de traza se basó en un análisis de duraciones de los eventos tipo LP Y VT registrados históricamente en el OVSP (Cadena, 2011). Se encontró que el 80 % de los sismos VT tienen duraciones menores que 21 segundos y el 90 % menor que 30 segundos. Para la sismicidad tipo LP el 90 % de los eventos tienen duración menor que 45 segundos y el 72 % menor que 30 segundos. Para la sismicidad tipo TRE el 3,7% de los eventos tienen una duración menor o igual a 51 segundos, y el 96,3% tiene duración mayor o igual a 51 segundos. De tal forma que, al tomar segmentos no mayores que 30 segundos se está evitando en la mayoría de los casos incluir información de ruido.

Teniendo en cuenta este análisis histórico, se construyeron y utilizaron 12 matrices de entrenamiento en las cuales el 50 % de los segmentos de traza corresponden a eventos tipo tectónico local de los años 2009 al 2013 y el otro 50 % se reparte de manera igual para

eventos VT de 2009 y 2010, LP de 2010 y TRE de 2010 y 2011.

5.2. Elección de las mejores topologías

Las matrices de entrenamiento se diferencian unas de otras tanto en el orden LPC en el que fueron generadas, pero también en los tiempos de corte de los segmentos de traza tal como lo indica la Tabla 5.1

Nombre de Matriz	Dimensión	Orden LPC	Tiempo de corte trazas
MATRIZ217	49x4367	7	21
MATRIZ219	63x4367	9	21
MATRIZ245	40x4224	5	24
MATRIZ247	56x4224	7	24
MATRIZ249	72x4224	9	24
MATRIZ275	45x3978	5	27
MATRIZ277	63x3159	7	27
MATRIZ279	81x3150	9	27
MATRIZ305	50x2848	5	30
MATRIZ307	70x2919	7	30
MATRIZ309	90x3195	9	30

Tabla 5.1: Configuración de matrices de entrenamiento. La cantidad de eventos tectónicos y volcánicos en cada matriz es de 50 % y 50 %.

Cada matriz fue normalizada en el intervalo $[-1,1]$ y se generaron con el modulo de pre-procesamiento de señales de *DSTYV*. Los objetivos de estas matrices fueron 1 para eventos tipo volcánicos y 0 para eventos tipo tectónicos locales.

El porcentaje de segmentos de traza utilizados para la validación variaban entre el 20 % y 30 % y el resto para el proceso de entrenamiento. En algunos entrenamientos se reserva un pequeño porcentaje para el error test pero este parámetro no proporciona resultados importantes en nuestro estudio. Las diferentes configuraciones de los parametros para generar las RNA se ven en la Tabla 5.2.

Se entrenaron 55 RNA pero se escogieron las mejores 40 respecto a su error de generalización, con una capa oculta, manteniendo el número de neuronas ocultas por debajo

del número de neuronas de entrada. Los valores de la tasa de aprendizaje y momento (momentum) se muestra en la tabla 5.2. El número máximo de épocas (iteración + validación) variaba entre 10000 y 50000 pero los entrenamientos terminaban alrededor de las 15000 épocas.

Nombre red	t. aprendizaje	Momento	% validación	% entrena- miento.	Topología red	E.global	E.generaliz.	orden lpc	duración segm. traza(s).
RED11277	0.4	0.4	25	75	63-13-1	0.1585	0.1712	7	27
RED6277	0.3	0.4	25	75	63-15-1	0.1571	0.1724	7	27
RED9277	0.4	0.2	25	75	63-13-1	0.1596	0.1726	7	27
RED10277	0.4	0.2	25	75	63-12-1	0.1627	0.1746	7	27
REDS5277	0.2	0	25	75	63-15-1	0.1528	0.1757	7	27
RED3277	0.2	0.1	25	75	63-6-1	0.1666	0.1798	7	27
RED4277	0.2	0	25	75	63-5-1	0.1867	0.1798	7	27
RED7277	0.3	0.4	25	75	63-14-1	0.1653	0.1805	7	27
RED2277	0.2	0	25	75	63-4-1	0.1697	0.1808	7	27
RED8277	0.3	0.2	25	75	63-14-1	0.1502	0.1812	7	27
RED4279	0.2	0.5	25	75	81-4-1	0.1791	0.1825	9	27
RED3279	0.2	0.4	25	75	81-5-1	0.1767	0.1835	9	27
RED6279	0.2	0.5	25	75	81-5-1	0.1732	0.1841	9	27
RED1277	0.2	0.4	20	80	63-5-1	0.1714	0.1848	7	27
REDS279	0.3	0.1	25	75	81-6-1	0.1721	0.1849	9	27
RED2217	0.2	0.4	25	70	49-17-1	0.1757	0.1854	7	21
RED1279	0.3	0.1	25	75	81-5-1	0.1799	0.1865	9	27
RED6275	0.5	0.5	25	75	45-4-1	0.1870	0.1869	5	27
RED2279	0.2	0.3	25	75	81-5-1	0.1729	0.1877	9	27
RED6307	0.2	0.5	25	75	70-18-1	0.1573	0.1878	7	30
REDS275	0.5	0.5	25	75	45-3-1	0.1861	0.1880	5	27
RED1247	0.2	0.4	30	70	56-17-1	0.1781	0.1887	7	24
RED1275	0.5	0.5	25	75	45-5-1	0.1763	0.1889	5	27
RED3217	0.5	0.4	20	65	49-17-1	0.1824	0.1890	7	21
REDS305	0.3	0.5	20	80	50-10-1	0.1726	0.1891	5	30
RED3245	0.2	0.4	20	80	40-5-1	0.1847	0.1895	5	24
RED4305	0.4	0.5	20	80	50-10-1	0.1890	0.1895	5	30
RED4307	0.2	0.6	25	75	70-25-1	0.1633	0.1897	7	30
RED4275	0.2	0.4	25	75	45-5-1	0.1886	0.1899	5	27
RED3305	0.6	0.5	20	80	50-10-1	0.1802	0.1904	5	30
RED2219	0.2	0.4	25	75	63-20-1	0.1764	0.1906	9	21
RED2275	0.5	0.5	25	75	45-8-1	0.1767	0.1913	5	27
RED1217	0.2	0.5	25	70	49-20-1	0.1700	0.1916	7	21
RED3275	0.2	0.4	25	75	45-6-1	0.1804	0.1919	5	27
RED3307	0.2	0.5	25	75	70-25-1	0.1559	0.1919	7	30
RED1219	0.3	0.4	25	75	63-20-1	0.1766	0.1920	9	21
RED7307	0.2	0.5	25	75	70-17-1	0.1797	0.1925	7	30
RED1307	0.5	0.5	25	75	70-10-1	0.1702	0.1930	7	30
RED2245	0.2	0.4	20	80	40-12-1	0.1742	0.1932	5	24
RED1309	0.5	0.6	25	75	90-23-1	0.1679	0.1932	9	30

Tabla 5.2: Valores de los errores global y de generalización para las RNA entrenadas. Esta tabla está ordenada de manera descendente de acuerdo al error de generalización, .

Enfocados en el error de generalización, los mejores errores se obtuvieron para las redes con las siguientes topologías: 63-13-1, 63-15-1, 63-12-1 y 63-6-1, (neuronas de entrada-neuronas ocultas-neuronas de salida), que corresponden a las redes entrenadas con la matriz *MATRIZ277*.

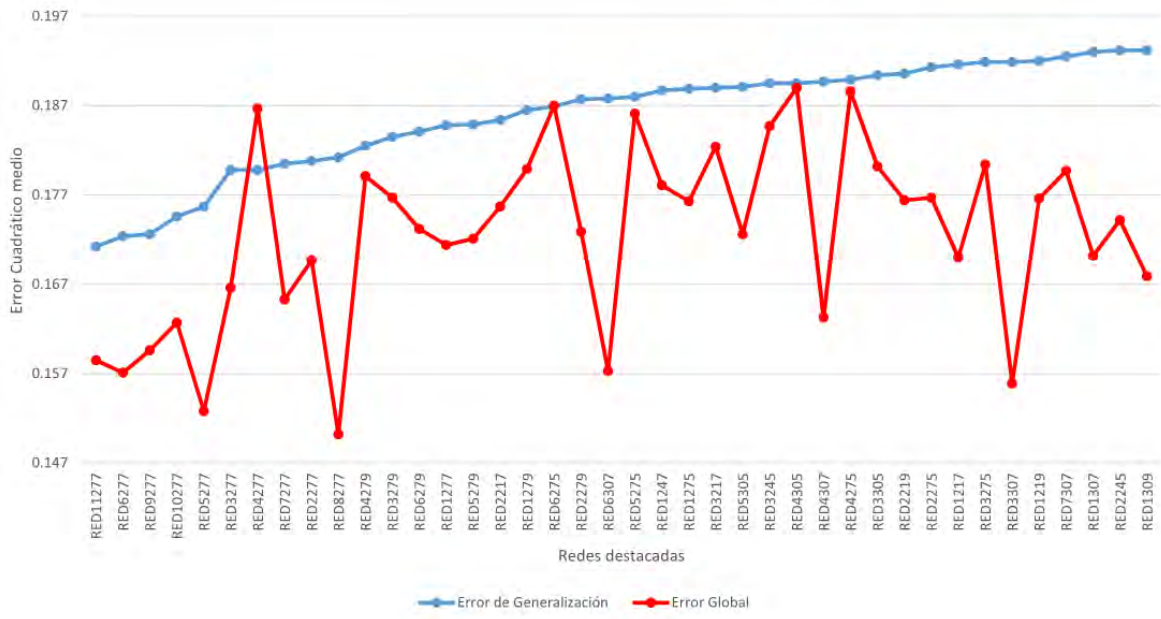


Figura 5.2: En esta gráfica se muestra el comportamiento de los mínimos errores globales y de generalización para las 40 RNA más destacadas. Las dos primeras topologías correspondientes a las RNA RED11277 (Figura 5.3) y RED6277 alcanzaron los valores mínimos en el error de generalización y tienen en común un momentum de 0,4.

Teniendo en cuenta que las RNA fueron seleccionadas por el criterio del menor error de generalización, también se tuvo presente que el error global no sea alto, que en el caso de los entrenamientos presentes en la Tabla 6.2 también fueron satisfactorios e inclusive siempre eran menores que los de generalización.

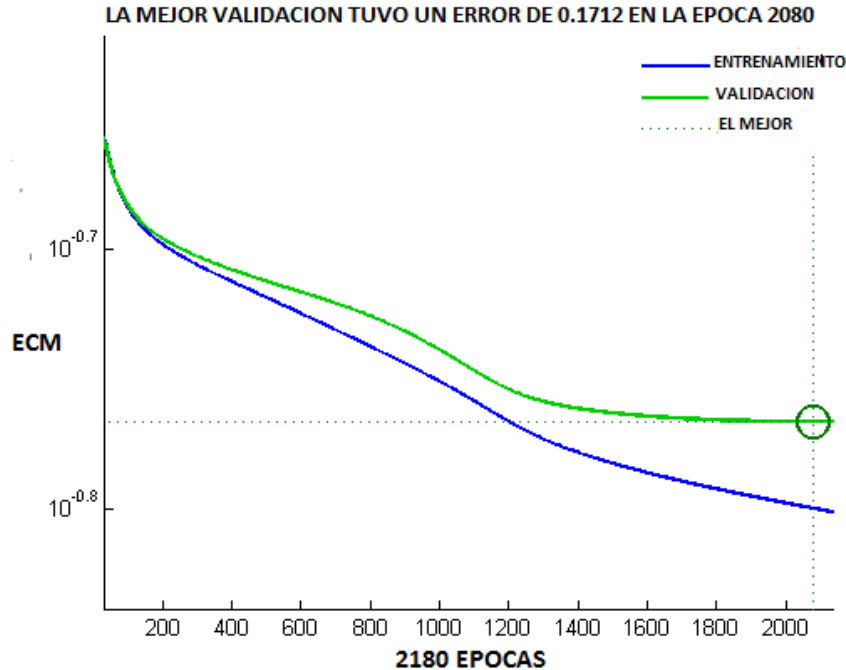


Figura 5.3: Variaciones del error global y de generalización con el número de épocas para el entrenamiento con RNA de topología 63-13-1. En esta gráfica se observa que el error global desciende hasta el final de las épocas, sin embargo el error de generalización empieza a incrementar.

5.3. Ejecución de las RNA

Se ejecutaron RNA con elementos que no se incluyeron en las matrices de entrenamiento, un total de 100: 20 eventos tipo LP, 15 tipo TRE, 15 tipo VT y 50 tipo tectónico local. Los resultados de estas ejecuciones de las RNA se muestran en la Tabla 5.3.

Los porcentajes de aciertos van desde 84 % a 100 % con un promedio del 93 % (Figura 5.4), los detalles de los resultados se pueden ver en la tabla 5.3. Al comparar los resultados de a tabla 5.2 con la tabla 5.3 se puede observar que la RNA que obtuvo el menor error de generalización (RED11277) no coincide con la de mayor acierto en la ejecución (RED3307). La RNA RED3307 fué la que alcanzó el 100 % de acierto, la cual tiene una topología 70-25-

1, fue entrenada con la matriz MATRIZ307, contiene 30 segundos de traza normalizada y 7 coeficientes de predicción por cada 300 muestras. Lo común entre las mejores RNA elegidas por el criterio de mínimo error de generalización (RED11277) y por el de mayor efectividad en ejecución (RED3307) es que fueron entrenadas con orden 7 LPC, de esta manera se puede ver que las RNA con 7 LPC dieron los mejores resultados.

Nombre red	% Acierto	Valor Mínimo	Valor Maximo	Promedio de abs(Objectivos-Salidas)	Desv. Estandar
RED3307	100	0.0111	0.10	0.0540	0.1134
RED9277	98	0.0062	0.64	0.3332	0.1418
RED5279	98	0.0132	0.91	0.4733	0.1841
RED5305	98	0.0010	0.96	0.4669	0.2105
RED4305	98	0.0096	1.05	0.4609	0.2279
RED2307	98	0.0082	1.26	0.5126	0.2349
RED11277	97	0.0033	0.51	0.3213	0.1265
RED1279	97	0.1279	0.92	0.4788	0.1603
RED7307	97	0.0190	1.26	0.5049	0.2385
RED2305	97	0.0412	1.05	0.4694	0.2188
RED6307	96	0.0019	1.10	0.5065	0.2499
RED4307	95	0.0101	1.18	0.4926	0.2388
RED3305	95	0.0038	1.01	0.4578	0.2127
RED6277	94	0.0025	0.64	0.3634	0.1450
RED5277	94	0.0033	0.71	0.3534	0.1505
RED8277	94	0.0025	0.70	0.3331	0.1546
RED7277	93	0.0127	0.87	0.3330	0.1572
RED2279	92	0.0774	0.88	0.4896	0.1608
RED1307	92	0.0315	1.31	0.4967	0.2576
RED5307	92	0.0039	1.14	0.4974	0.2378
RED4279	91	0.0211	0.93	0.4821	0.1763
RED1217	91	0.0450	1.16	0.4980	0.1951
RED4277	90	0.0041	0.69	0.3576	0.1456
RED3279	90	0.0531	0.96	0.4782	0.1757
RED1305	90	0.0673	1.00	0.4622	0.2159
RED10277	86	0.0209	0.75	0.3309	0.1485
RED3277	86	0.0014	0.72	0.3424	0.1430
RED2309	86	0.0080	1.13	0.4591	0.2315
RED1277	85	0.0102	0.80	0.3371	0.1564
RED2277	84	0.0337	0.71	0.3429	0.1509

Tabla 5.3: Resultados de la ejecución de las mejores 30 de las 55 RNA entrenadas para la clasificación entre los tipos volcánicos y tectónicos.



Figura 5.4: Porcentajes de acierto vs cada RNA utilizada en el proceso de ejecución. Las RNA con LPC 7 dan los mejores resultados para discriminar entre sismicidad tectónica y volcánica.

Capítulo 6

Discusión

La técnica de usar RNA con entrenamiento supervisado como sistemas clasificadores es una alternativa viable para obtener buenos resultados, pero esta técnica se ve limitada únicamente a basarse en los ejemplos suministrados por el usuario, de tal manera que, una parte importante al momento de crear una RNA es tener los ejemplos lo suficientemente adecuados para que al momento de clasificar eventos desconocidos, la RNA funcione correctamente. Hay que resaltar que para este trabajo se utilizaron resultados de la componente vertical una sola estación (**Estación Anganoy ANGV**) ya que cada estación proporciona una respuesta espectral distinta para un mismo evento y también para obtener resultados de un mismo instrumento para los diferentes tipos de sismos.

Para mejorar el método utilizado en este estudio se podrían usar eventos no solo fuera de línea sino que la discriminación pudiese realizarse en tiempo real, lo que conllevaría a utilizar otras técnicas de análisis espectral que permitan representar los eventos de manera precisa, pero que a la vez sean adecuados a la hora de entrenar las RNA.

En la fase de entrenamiento de las RNA se obtuvieron las mejores por criterio de mínimo error de generalización, pero hubieron redes que no fueron las mejores bajo este criterio que alcanzaron los mejores resultados en la ejecución con datos desconocidos, mostrando así que el parámetro de error de generalización es muy importante pero no es el único que se debe tener en cuenta a la hora de escoger la mejor RNA.

Para la fase de ejecución de RNA el método utilizado funciona muy bien, logrando en las pruebas realizadas aciertos superiores al 90 % en su mayoría, con un promedio del 93 % y en uno de los casos se obtuvo un acierto del 100 %, haciendo de este estudio lo suficientemente confiable como para utilizarlo como clasificador de sismicidad tectónica y volcánica.

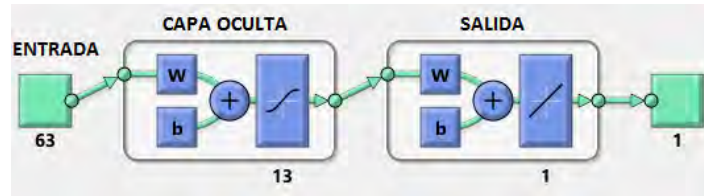


Figura 6.1: Topología de la mejor RNA respecto a mínimo error de generalización.

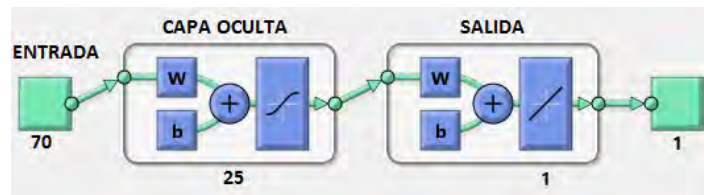


Figura 6.2: Topología de la mejor RNA respecto a aciertos frente a datos desconocidos.

El programa utilizado en este estudio *DSTYV*, da la libertad al usuario de manipular a su gusto muchos de los parámetros de entrenamiento de las RNA haciendo así, mas personalizada la creación de las redes. Por medio de ensayo-error se fueron escogiendo minuciosamente las mejores configuraciones.

Los resultados obtenidos indican que esta técnica de discriminación de sismicidad puede ser una herramienta de ayuda, pero, es necesario dejar claros algunos aspectos acerca de la capacidad y manejo del presente trabajo:

- Los resultados del sistema discriminador deben ser supervisados por un temático ya que el programa generado en este estudio no garantiza en todos los casos un error adecuado. Además se debe realizar revisiones y pruebas preliminares de la herramienta computacional antes de su utilización.
- Es de vital importancia la correcta elección de los ejemplos de entrenamiento para que la herramienta computacional pueda realizar el trabajo deseado, en caso contrario, podrían surgir errores en la ejecución del programa. Por tal razón se deben evitar las señales relacionadas con ruidos y con eventos ajenos a los que puede clasificar el sistema.

Capítulo 7

Conclusiones y Recomendaciones

7.1. Conclusiones

La aplicación de la metodología propuesta en este estudio proporciona un alto grado de efectividad y confianza en cuanto a la discriminación de sismicidad tectónica y volcánica. Las trazas pre-procesadas representan correctamente las características de las señales sísmicas y los porcentajes de acierto son muy altos para la mayoría de las RNA entrenadas.

La discriminación de sismicidad tectónica y volcánica alcanzó en promedio para las redes entrenadas el 93 % de aciertos y en uno de los casos se presenta una RNA con 100 % de acierto. Gracias a la flexibilidad del programa utilizado en este estudio fue posible manipular de muchas maneras los parámetros de entrenamiento y por tal razón se generaron diversas configuraciones de entrenamiento.

Respecto al orden de los LPC, las matrices generadas con 7 coeficientes de predicción lineal por cada 300 muestras con topologías cuyas neuronas en la capa oculta están en el rango entre 10 y 15 y con tasas de aprendizaje alrededor de 0.4 mostraron los mejores resultados.

El programa diseñado en este estudio *DSTYV*, mostró a nivel operativo el desempeño esperado, cumpliendo correctamente con las expectativas planteadas al generarlo, como la de pre-procesar matrices, entrenar RNA y posteriormente ejecutarlas, además, su interfaz da la libertad de cambiar muchos de los parámetros de entrenamiento.

Los datos implementados, los cuales son la materia prima de este trabajo, fueron los suficientes para organizar las diferentes matrices tanto de entrenamiento como de prueba de las RNA.

7.2. Recomendaciones

La utilización de RNA para la discriminación de sismicidad tectónica y volcánica puede extenderse a cualquier tipo de estación de la red sísmica, pero para ello, es necesario generar inicialmente las matrices de entrenamiento, validación y prueba de cada estación a estudiar.

Se sugiere explorar otras técnicas de representación y reducción de la dimensión de las señales, las cuales podrían combinarse la técnica LPC y así obtener buenos resultados incluso extrayendo características en otros dominios.

Se recomienda realizar estudios de la discriminación de sismicidad no únicamente relacionada con la volcánica o tectónica, sino que también incluya en su clasificación señales inusuales tales como las atribuidas a tormentas eléctricas, vientos entre otras similares, y teniendo datos de estos tipos, conformar las matrices pre-procesadas, de validación y de prueba y así construir RNA capaces de discriminar entre estas clases de eventos.

Apéndice A

Código del programa

A continuación se muestra el código de los tres módulos de *DSTYV* para la obtención de las RNA en este trabajo:

PRE-PROCESAMIENTO DE DATOS

```
1 %EXAMINAR CARPETAS
2 function examinar_Callback(hObject , eventdata , handles)
3 %SELECCIONAR LOS DATOS LOS CUALES VAN A CONFORMAR LA MATRIZ
4 handles.A=uigetfile ('*.asc','MultiSelect','on');
5 %ES EL NUMERO DE ARCHIVOS SELECCIONADOS
6 handles.we=length(handles.A);
7 set(handles.casillax,'string',handles.A);
8 guidata(hObject,handles);
9
10 function casillai2_Callback(hObject , eventdata , handles)
11 Vali2=get(hObject,'String');
12 handles.casillai2=Vali2;
13 guidata(hObject,handles);
14
15 function edit14_Callback(hObject , eventdata , handles)
16 valida=get(hObject,'String');
17 validac=str2double(valida);
18 handles.validacion=validac;
19 entradasc=strcat(handles.entrada);
20 c2=size(entradasc,2);
```

```
21 handles.mn=0.01*handles.validacion*c2;
22 handles.mnr=round(handles.mn);
23 set(handles.casillai4,'string',handles.mnr);
24 guidata(hObject,handles);
25
26
27 function entrenamiento_Callback(hObject,eventdata,handles)
28 entrena=get(hObject,'String');
29 entrenamien=str2double(entrena);
30 handles.entrenamiento=entrenamien;
31 guidata(hObject,handles);
32
33 function tcorte_Callback(hObject,eventdata,handles)
34 tcort=get(hObject,'String');
35 tc=str2double(tcort);
36 handles.tcorte=tc;
37 guidata(hObject,handles);
38
39 %GENERA MATRIZ DE ENTRENAMIENTO
40 function gmatriz_Callback(hObject,eventdata,handles)
41 bp=waitbar(0,'Generando Matriz...')
42 for k=1:handles.we
43     m{k}=importdata(handles.A{k}); %IMPORTA LOS DATOS
44     B{k}=m{k}.textdata; %SOLO UNICAMENTE TOMA LOS DATOS NUMERICOS
45     s{k}=str2double(B{k}); %TOMA LOS DATOS NUMERICOS Y LOS VUELVE
46     %COMPATIBLES PARA VOLVERLOS MATRIZ
47     sr{k}=s{k}(6:(handles.tcorte*100)+5); %CORTA ENCABEZADOS
48     matrizdatos1=[sr{1:end}]; %MATRIZ DE LOS DATOS
49     mn=mean(matrizdatos1); %VALOR MEDIO COLUMNAS "matrizdatos1"
50     matdatos1(:,k)=matrizdatos1(:,k)-mn(1,k); %MATRIZ DE DATOS ←
    SIN OFFSET
```

```
51     [b1 , a1]= butter (3 ,[0.05 ,0.3] );
52     matrizdatos=filter (b1 , a1 , matdatos1); %FILTRO PASABANDA DE 5 A↔
        30 Hz
53     handles . matrizdatos=matrizdatos ;
54     waitbar (k/ handles . we , bp)
55 end
56 close (bp)
57 uisave ( 'matrizdatos ' , 'matriz_datos ' );
58 guidata ( hObject , handles );
59
60 function pushbutton17_Callback(hObject , eventdata , handles)
61 bp1=waitbar(0, 'Abriendo Matriz ... ');
62 delete( 'matriz de datos.xlsx ' )
63 xlswrite( 'matriz de datos.xlsx ' , handles . matrizdatos )
64 winopen( 'matriz de datos.xlsx ' )
65     waitbar (1 , bp1)
66 close (bp1)
67 guidata ( hObject , handles );
68
69 function tcorte_CreateFcn(hObject , eventdata , handles)
70 if ispc && isequal( get(hObject , 'BackgroundColor ' ) , get(0 , '↔
        defaultUicontrolBackgroundColor ' ))
71     set ( hObject , 'BackgroundColor ' , 'white ' );
72 end
73
74 %ORDEN LPC
75 function ordenlpc_Callback(hObject , eventdata , handles)
76 olpc=get( hObject , 'String ' );
77 ol=str2double( olpc );
78 handles . ordenlpc=ol ;
79 guidata ( hObject , handles );
```

```

80
81 function ordenlpc_CreateFcn(hObject , eventdata , handles)
82 if ispc && isequal(get(hObject , 'BackgroundColor') , get(0 , '↵
    defaultUicontrolBackgroundColor'))
83     set(hObject , 'BackgroundColor' , 'white');
84 end
85
86 %GENERA MATRIZ DE LPC.
87 function gmatrizlpcs_Callback(hObject , eventdata , handles)
88 bp1=waitbar(0 , 'Generando Matriz de LPCs... ')
89 for k=1:handles.we
90     m{k}=importdata(handles.A{k}); %IMPORTA LOS DATOS
91     B{k}=m{k}.textdata;           %SOLO UNICAMENTE TOMA LOS ↵
        DATOS NUMERICOS
92     s{k}=str2double(B{k}); %TOMA LOS DATOS NUMERICOS Y LOS VUELVE
93     %COMPATIBLES PARA VOLVERLOS MATRIZ
94     sr{k}=s{k}(6:(handles.tcorte*100)+5); %CORTA ENCABEZADOS
95     matrizdatos1=[sr{1:end)];           %MATRIZ DE LOS DATOS
96     mn=mean(matrizdatos1); %VALOR MEDIO DE CADA COLUMNA DE "↵
        matrizdatos1"
97     matdatos1(:,k)=matrizdatos1(:,k)-mn(1,k); %MATRIZ DE DATOS ↵
        SIN OFFSET
98     [b1 , a1]=butter(3 , [0.05 , 0.3]);
99     matdatos=filter(b1 , a1 , matdatos1); %FILTRO PASABANDA ENTRE 5 Y↵
        30 Hz
100    L2=reshape(matdatos(:,k) , 300 , length(matdatos(:,k))/300); %↵
        TRANSFORMA
101    %CADA COLUMNA DE matdatos Y LO TRANSFORMA EN UNA MATRIZ 300x(↵
        long de vector/300)
102    L=normc(L2); %NORMALIZA COLUMNA
103    ncolumns=handles.tcorte*100/300;

```

```
104
105     for z=1:ncolumnas
106         ventaneo=hanning(300);      %VECTOR HAMMING PARA 300 ←
            MUESTRAS
107         L1(:,z)=L(:,z).*ventaneo; %SE HACE VENTANEO A CADA COLUMNA ←
            DE
108         %LAS MATRICES L
109         coef1=lpc(L1,handles.ordenlpc);
110         coef=coef1';      %cada columna son los lpc de cada columna ←
            de L
111     coef(1,:)=[];      %QUITA LOS UNOS
112     end
113     ve(:,k)=reshape(coef,(handles.ordenlpc)*ncolumnas,1); %MATRIZ←
            DE LPC'S
114     %DE CADA 300 MUESTRAS DE LAS COLUMNAS DE LA MATRIZ matdatos
115     matriz_preprocesada=ve;
116     handles.sixq=length('m');
117     assignin('base','ndatos',handles.sixq)
118     handles.matriz_preprocesada=matriz_preprocesada;
119     waitbar(k/handles.we,bp1); %Barra de progreso
120 end
121     close(bp1);
122     uisave('matriz_preprocesada','matriz_preprocesada');
123     guidata(hObject,handles);
124
125 function pushbutton18_Callback(hObject,eventdata,handles)
126 bp2=waitbar(0,'Abriendo Matriz de LPCs...');
127 delete('matriz_preprocesada.xlsx')
128 xlswrite('matriz_preprocesada.xlsx',handles.matriz_preprocesada)
129 winopen('matriz_preprocesada.xlsx')
130 waitbar(1,bp2)
```

```

131 close (bp2)
132 guidata (hObject , handles );

```

ENTRENAMIENTO DE RNA

```

1
2 function casillaent_Callback (hObject , eventdata , handles)
3 Valent=get (hObject , 'String ');
4 handles . casillaent=Valent;
5 guidata (hObject , handles );
6
7 %—— Executes on button press in examinar2.
8 function examinar2_Callback (hObject , eventdata , handles)
9 handles . entradas=uigetfile ( '*.mat' ); %SIRVE PARA SELECCIONAR LOS
10 %DATOS LOS CUALES VAN A CONFORMAR LA MATRIZ
11 handles . entrada=importdata ( handles . entradas );
12 entradasc=strcat ( handles . entrada );
13 c2=size (entradasc ,2);
14 c1=size (entradasc ,1);
15 [ubi2 , handles . nombreent , extens2]= fileparts ( handles . entradas )
16 set ( handles . casillai3 , 'string ' ,c1);
17 set ( handles . casillai1 , 'string ' ,c2);
18 set ( handles . casillaent , 'string ' , handles . nombreent );
19 assignin ( 'base ' , 'entradas ' , handles . entrada )
20 handles . we1=length ( handles . entrada ) %ES EL NUMERO DE ARCHIVOS ←
    SELECCIONADOS
21 guidata (hObject , handles );
22
23 function casillaobj_Callback (hObject , eventdata , handles)
24 Valobj=get (hObject , 'String ');
25 handles . casillaobj=Valobj;

```

```
26 guidata(hObject, handles);
27
28 function examinar3_Callback(hObject, eventdata, handles)
29 handles.objetivos=uigetfile('*.mat'); %SIRVE PARA SELECCIONAR LOS↔
    DATOS
30 %LOS CUALES VAN A CONFORMAR LA MATRIZ
31 handles.objetivo=importdata(handles.objetivos);
32 [ubi3, handles.nombreobj, extens3]= fileparts(handles.objetivos)
33 set(handles.casillaobj, 'string', handles.nombreobj);
34 assignin('base', 'objetivos', handles.objetivo)
35 handles.we2=length(handles.objetivo) %ES EL NUMERO DE ARCHIVOS ↔
    SELECCIONADOS
36 guidata(hObject, handles);
37
38 function edit18_Callback(hObject, eventdata, handles)
39 e18=get((hObject, 'String');
40 ed18=str2double(e18);
41 handles.edit18=ed18;
42 guidata(hObject, handles);
43
44 function nco_Callback(hObject, eventdata, handles)
45 nunc=get((hObject, 'String');
46 nnco=str2double(nunc);
47 handles.nco=nnco;
48 set(handles.edit18, 'string', handles.nco);
49 guidata(hObject, handles);
50
51 function nco_CreateFcn(hObject, eventdata, handles)
52 if ispc && isequal(get( hObject, 'BackgroundColor'), get(0, '↔
    defaultUicontrolBackgroundColor'))
53     set( hObject, 'BackgroundColor', 'white');
```

```
54 end
55
56 function edit20_Callback(hObject, eventdata, handles)
57 e20=get(hObject, 'String');
58 ed20=str2double(e20);
59 handles.edit20=ed20;
60 guidata(hObject, handles);
61
62 function ta_Callback(hObject, eventdata, handles)
63 tasa=get(hObject, 'String');
64 tasaa=str2double(tasa);
65 handles.ta=tasaa;
66 set(handles.edit20, 'string', handles.ta);
67 guidata(hObject, handles);
68
69 function ta_CreateFcn(hObject, eventdata, handles)
70 if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, '↵
    defaultUicontrolBackgroundColor'))
71     set(hObject, 'BackgroundColor', 'white');
72 end
73
74 function edit21_Callback(hObject, eventdata, handles)
75 e21=get(hObject, 'String');
76 ed21=str2double(e21);
77 handles.edit21=ed21;
78 guidata(hObject, handles);
79
80 function edit13_Callback(hObject, eventdata, handles)
81 mome=get(hObject, 'String');
82 momento=str2double(mome);
83 handles.moment=momento;
```



```
84 set(handles.edit21,'string',handles.moment);
85 guidata(hObject,handles);
86
87 function edit22_Callback(hObject,eventdata,handles)
88 e22=get(hObject,'String');
89 ed22=str2double(e22);
90 handles.edit22=ed22;
91 guidata(hObject,handles);
92
93 function ite_Callback(hObject,eventdata,handles)
94 iter=get(hObject,'String');
95 itera=str2double(iter);
96 handles.ite=itera;
97 set(handles.edit22,'string',handles.ite);
98 guidata(hObject,handles);
99
100 function ite_CreateFcn(hObject,eventdata,handles)
101 if ispc && isequal(get(hObject,'BackgroundColor'),get(0,'↵
    defaultUicontrolBackgroundColor'))
102     set(hObject,'BackgroundColor','white');
103 end
104
105 %PARAMETROS DE ENTRENAMIENTO
106 function entr_Callback(hObject,eventdata,handles)
107 handles.net=feedforwardnet(handles.nco,'traingdm'); %CREA LA RED ↵
    ,CUYA CAPA
108 %CULTA TIENE 20 NEURONAS CUYA FUNCION DE ACTIVACION ES LA ↵
    FUNCION SIGMOIDEA%
109 handles.net = configure(handles.net,handles.entrada,handles.↵
    objetivo);
110 handles.net.trainParam.lr=handles.ta; %TASA DE APRENDIZAJE%
```

```
111 %TAMAÑO DE PASOS PARA ENCONTRAR MINIMO LOCAL%
112 handles.net.divideParam.trainRatio=handles.entrenamiento/100;
113 handles.net.divideParam.valRatio=handles.validacion/100; %↵
    porcentaje de
114 %ejemplos para validación
115 handles.net.divideParam.testRatio=1-handles.net.divideParam.↵
    valRatio-handles.net.divideParam.trainRatio;
116 handles.net.trainParam.mc=handles.moment; %configura el ↵
    parámetros momento
117 handles.net.trainParam.max_fail=100;
118 handles.net.trainParam.epochs=handles.ite; %NUMERO DE ITERACIONES%
119
120 %%ENTRENA LA RED%%
121 [handles.net,handles.tr] = train(handles.net,handles.entrada,↵
    handles.objetivo);
122 handles.salida=handles.net(handles.entrada);
123 assignin('base','hsalida',handles.salida)
124 assignin('base','htrain',handles.tr)
125 nntraintool('close')
126 guidata(hObject,handles);
127
128 function pushbutton16_Callback(hObject,eventdata,handles)
129 view(handles.net)
130 guidata(hObject,handles);
131
132 function pushbutton6_Callback(hObject,eventdata,handles)
133 figure(1);
134 plottrainstate(handles.tr) %ERROR CUADRATICO MEDIO VS Iteraciones↵
    %
135 guidata(hObject,handles);
136
```

```
137 %GRAFICAS DE ENTRENAMIENTO
138 function pushbutton7_Callback(hObject, eventdata, handles)
139 figure(2);
140 trOut = handles.salida(handles.tr.trainInd);
141 vOut = handles.salida(handles.tr.valInd);
142 tsOut = handles.salida(handles.tr.testInd);
143 trTarg = handles.objetivo(handles.tr.trainInd);
144 vTarg = handles.objetivo(handles.tr.valInd);
145 tsTarg = handles.objetivo(handles.tr.testInd);
146 e1 = vTarg - vOut;
147 e3 = handles.objetivo - handles.salida;
148 ploterrhist(e1, 'Validación', e3, 'Entrenamiento')
149 guidata(hObject, handles);
150
151 function edit23_Callback(hObject, eventdata, handles)
152 e23=get(hObject, 'String');
153 ed23=str2double(e23);
154 handles.edit23=ed23;
155 guidata(hObject, handles);
156
157 function edit10_Callback(hObject, eventdata, handles)
158 nomred=get(hObject, 'String');
159 handles.namered=nomred;
160 set(handles.edit23, 'string', handles.namered);
161 guidata(hObject, handles);
162
163 %GUARDAR RED
164 function pushbutton8_Callback(hObject, eventdata, handles)
165 genFunction(handles.net, handles.namered);
166 save(handles.namered)
167 load(handles.namered)
```

```
168     msgbox('Red guardada en carpeta raíz del programa','Mensaje')
169     guidata(hObject,handles);
170
171     function casilla_Callback(hObject, eventdata, handles)
172     Val=get(hObject, 'String');
173     handles.casilla=Val;
174     guidata(hObject,handles);
```

EJECUCIÓN DE RNA

```
1 function pushbutton10_Callback(hObject, eventdata, handles)
2 %OPCION DE SELECCIONAR RED DE TRABAJO
3 [handles.redneuro]=uigetfile('*.m','Seleccione la red de trabajo'↵
4     );
5 [ubi,handles.nombred,extens]= fileparts(handles.redneuro)
6 set(handles.casilla, 'String',handles.nombred);
7 guidata(hObject,handles);
8
9 function casilla1_Callback(hObject, eventdata, handles)
10 Val1=get(hObject, 'String');
11 handles.casilla1=Val1;
12 guidata(hObject,handles);
13
14 function casilla1_CreateFcn(hObject, eventdata, handles)
15
16 %OPCION DE SELECCIONAR LA MATRIZ A DISCRIMINAR SUS DATOS
17 function pushbutton11_Callback(hObject, eventdata, handles)
18 handles.datosentr = uigetfile('*.mat','Seleccione el archivo .txt↵
19     ');
20 handles.l=importdata(handles.datosentr);
21 [ubil,handles.nombrema,extens1]= fileparts(handles.datosentr);
```

```

20 set(handles.casilla1, 'string', handles.nombrema);
21 guidata(hObject, handles);
22
23 function pushbutton12_Callback(hObject, eventdata, handles)
24
25 function pushbutton13_Callback(hObject, eventdata, handles)
26 handles.redneuro=num2str(handles.redneuro);
27 handles.datosentr=num2str(handles.datosentr);
28 assignin('base', 'Entradas', handles.l);
29 handles.AQ=feval(handles.nombred, handles.l);
30 fc=round(handles.AQ);
31 assignin('base', 'Salidas', handles.AQ)
32 msgbox('La red se ha ejecutado correctamente', 'Mensaje')
33 guidata(hObject, handles);

```

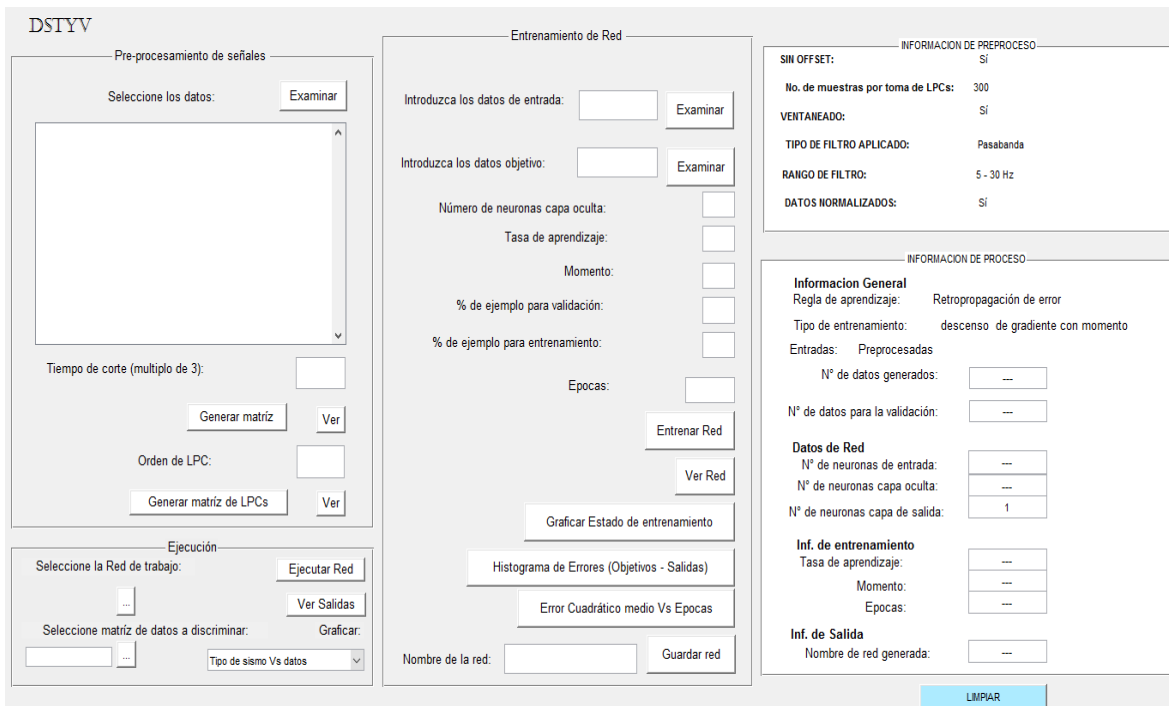


Figura A.1: Entorno gráfico DSTYV.

Bibliografía

- [1] M. CALVACHE, R. CÁRDENAS, G. CORTÉS, A. HURTADO, C. LAVERDE, A. RUALES O. SUAREZ. *Mapa de Amenaza Volcánica del Galeras. Publicación especial de Ingeominas, 1998.*
- [2] CADENA, OSCAR. DIMATÉ, CRISTINA. *Detección y clasificación automática de registros sísmicos en el Observatorio Vulcanológico y Sismológico de Pasto utilizando redes neuronales artificiales 2011.*
- [3] BENBRAHIM, MOHAMMED. DAOUDI, ADIL. BENJELLOUN, KHALID. IBENBRAHIM, AOMAR. *Discrimination of seismic signals using artificial neuronal networks; Enformatika v4 2005 ISSN 1305-5313; Morocco, Unite Estates.*
- [4] BANCHS, RAFAEL. JIMENEZ, JUAN. VALENCIANO, ALEJANDRO. MARQUÉZ, DAVID. *Clasificación de datos geofísicos mediante el uso de atributos sísmicos no convencionales y redes neuronales. 2002.*
- [5] LATTEK, J. H. *Volcanological observations at Tongariro National Park, 2. Types and classification of volcanic earthquakes.* Department of Scientific and Industrial Research, Geophysics Division, Wellington, New Zealand, 1979.
- [6] CHOUET, B. *Long-period volcano seismicity: its source and use in eruption forecasting*, 1981.
- [7] CHOUET, B. *Excitation of a buried magmatic pipe: a seismic source model for volcanic tremor. Journal of Geophysical Research, 1985.*
- [8] CHOUET, B. *Dynamics of a fluid-driven crack in three dimensions by defined difference method. Journal of Geophysical Research, 1986.*
- [9] CHOUET, B. *Resonance of a fluid-driven crack: radiation properties and implications*

- for the source of long-period events and harmonic tremor. Journal of Geophysical Research, 1988.*
- [10] CHOUET, B. *A seismic model for the source of long-period events and harmonic tremor. In: P. Gasparini, R. Scarpa and K. Aki (Editors), Volcanic Seismology. Springer-Verlag, Berlin, 1992.*
- [11] CHOUET, B. *Long-period volcano seismicity: its source and use in eruption forecasting, 1996.*
- [12] GOMEZ, D. M., TORRES, R. A., SEIDL, D., HELLEG, M., RADEMECHER, H. *Tornillos seismic events at Galeras volcano, Colombia: Asummary and new information from broad band three-component measurements . Ann. Geofis, 1999.*
- [13] DEL PEZZO, EDOARDO. ESPOSITO, ANNA. GIUDICE, FLORA. MARINARO, MARIA.MARTINI, MARCELLO. SCARPETTA, SILVIA *Discrimination of earthquakes and underwater explosions using neuronal networks., 2002.*
- [14] DEL BRIO, BONIFACIO. SANZ,ALFREDO. *Redes Neuronales y Sistemas Difusos; Universidad de Zaragoza. 2002.*
- [15] GOMEZ, D. M., TORRES, R. A., SEIDL, D., HELLEG, M., RADEMECHER, H. *Tornillos seismic events at Galeras volcano, Colombia: Asummary and new information from broad band tree-component measurements, 1999.*
- [16] GOMEZ, D. M., TORRES, R. A. *Unusual low-frequency volcanic seismic events slowly decaying coda waves observed at Galeras and other volcanoes. J. Volcanol. Geoth. Res, 1997.*
- [17] MALONE, S.D. *Volcanic earthquakes:Examples from Mount St. Helnes, in Earthquake, Theory and Interpretation, edited by H. Kanamori and E. Boshci, Soc. IIta. Di Fisica, pp. 436-455, Bologna, Italy, 1983.*

-
- [18] MINAKAMI, T. *Seismology of volcanoes in Japan*. In: L. Civetta, P. Gasparini, G. Luongo and A. Rapolla (Editors), *Physical Volcanology*, Elsevier Scientific Publishing Company. The Netherlands, pp. 1-27, 1974.
- [19] PROAKIS, JOHN. MANOLAKIS, DIMITRIS. *Tratamiento digital de señales*. 1998.
- [20] RUGELES, ALVARO. CADENA OSCAR. *Discriminación entre registro sísmicos tipo LP y VT producidos por el volcán Galeras, Utilizando Redes Neuronales Artificiales*. Universidad de Nariño, 2006.
- [21] SAKURABA, ATARU. OIKAWA, JUN. IMANISHI, YUICHI. *Free oscillations of a fluid sphere in an infinite elastic medium and long-period volcanic earthquakes*. Earth plane space, 2002.
- [22] BISHOP, C. *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, 1995.
- [23] GOMEZ DIEGO. LAVERDE CARLOS. NARVAEZ LOURDES. ORTEGA ADRIANA. SILVA BETTY. TORRES JESÚS. TORRES ROBERTO. *Catálogo de señales sísmicas volcánicas de Colombia*. INGEOMINAS. 2004.