

**DISCRIMINACION DE REGISTROS
SISMICOS TIPO LPS, VTA Y HBD
ORIGINADOS POR EL VOLCAN GALERAS
UTILIZANDO REDES NEURONALES
ARTIFICIALES**

OSCAR ERNESTO CADENA IBARRA

Trabajo presentado como requisito previo
para optar el Título de Físico

Director
Dr. Alvaro Rugeles Pérez

UNIVERSIDAD DE NARIÑO
FACULTAD DE CIENCIAS NATURALES Y MATEMÁTICAS
DEPARTAMENTO DE FÍSICA
SAN JUAN DE PASTO
NOVIEMBRE 2006

RESUMEN

La acertada interpretación del fenómeno volcánico depende en gran medida de la correcta clasificación de los registros asociados a los diferentes tipos de eventos sísmicos. En este trabajo los registros sísmicos asociados a la actividad del volcán Galeras, son primero clasificados o discriminados entre los tipos LPS y VTA y luego, entre los tipos LPS y HBD. Para lograr esta clasificación se han implementado tres etapas: extracción de los datos, reducción de su dimensionalidad y uso de una Red Neuronal Artificial (RNA). La primera etapa requiere la elección de los registros a tratar; se escogieron 2500 en total y su formato original fue convertido a uno que el programa MATLAB 6.0 acepta. En la segunda etapa, aplicando la técnica llamada Coeficientes de Predicción Lineal (LPC), los registros se representan en forma compacta. La tercera etapa trata del uso de un Perceptrón Multicapa para la tarea de clasificación. El procedimiento descrito en este trabajo podría ser generalizado a la clasificación de otros tipos de eventos sísmicos con buenos resultados en su aplicación.

ABSTRAC

The certain interpretation of the volcano phenomena depends on a great measurement of the correct classification of the registers associated to different kind of seismic events. In this work the seismic registers associated to the Galeras Volcano activity, are first classified or discriminated between the kinds LPS and VTA, afterwards between the kinds LPS and HBD. To can make this classification has been implemented 3 levels: data extraction, dimensionality reduction and the use of a Artificial Neuronal Networks (RNA). The first level requires the election of the registers that are treated; are chosen in total 2500 and its original format was converted to one that MATLAB 6.0 program accepts. In the second level, applying the technique called Linear Prediction Coefficients (LPC), the registers are represented in a compact form. Finally the third level treat about the use of a Multilayer Perceptron to can make the classification work. The described procedure in this work could be generalized to the classification of other types of seismic events with good results in its application.

Contenido

| | |
|---|------------|
| Introducción | VII |
| OBJETIVOS | IX |
| 0.1. Objetivos Generales | IX |
| 0.2. Objetivos Específicos | IX |
| ESTADO DEL ARTE | X |
| 1. COEFICIENTES DE PREDICCIÓN LINEAL (LPC) | 1 |
| 1.1. Estimación de los LPC | 2 |
| 1.2. Método de autocorrelación | 3 |
| 1.3. Procedimiento de Levinson Durbing | 4 |
| 2. REDES NEURONALES ARTIFICIALES (RNA) | 6 |
| 2.1. La neurona artificial | 6 |
| 2.2. Redes Neuronales Artificiales | 8 |
| 2.2.1. Definición de red neuronal | 8 |
| 2.2.2. Modos de operación: recuerdo y aprendizaje | 9 |
| 2.2.3. Fase de aprendizaje | 9 |
| 2.2.4. Fase de recuerdo o ejecución: estabilidad. | 10 |
| 2.2.5. El Perceptrón Multicapa (MLP) | 10 |
| 3. METODOLOGIA | 12 |
| 3.1. DATOS | 12 |
| 3.1.1. Elección y extracción de las trazas | 13 |
| 3.2. REPRESENTACIÓN Y REDUCCIÓN DE DIMENSIONALIDAD | 15 |
| 3.2.1. Elección del tamaño de los intervalos de registro. | 16 |
| 3.2.2. Elección del número de LPC para cada segmento | 16 |
| 3.3. CLASIFICACION | 18 |
| 4. RESULTADOS | 24 |
| 4.1. Clasificación 1 | 24 |
| 4.2. Clasificación 2 | 29 |

| | |
|--------------------------------|-----------|
| 4.3. Clasificación 3 | 34 |
| CONCLUSIONES | 39 |
| RECOMENDACIONES | 40 |
| BIBLIOGRAFIA | 41 |
| ANEXOS | 42 |

Lista de tablas

| | |
|--|----|
| 4.1. Resultados lote de prueba 1.1 | 25 |
| 4.2. Resultados lote de prueba 1.2 | 25 |
| 4.3. Resultados lote de prueba 1.3 | 25 |
| 4.4. Mejores Configuraciones para clasificación 1 | 29 |
| 4.5. Resultados lote de prueba 2.1 | 30 |
| 4.6. Resultados lote de prueba 2.2 | 30 |
| 4.7. Resultados lote de prueba 2.3 | 30 |
| 4.8. Mejores Configuraciones para clasificación 2 | 34 |
| 4.9. Resultados lote de prueba 3.1 | 34 |
| 4.10. Resultados lote de prueba 3.2 | 36 |
| 4.11. Resultados lote de prueba 3.3 | 36 |
| 4.12. Mejores Configuraciones para clasificación 3 | 36 |

Lista de Figuras

| | |
|--|----|
| 2.1. Modelo de Neurona Estandar (Tomada del Tutorial de MATLAB) | 7 |
| 2.2. Gráficas de información acerca del proceso de entrenamiento | 7 |
| 2.3. Capa de una Red Neuronal Artificial (Tomada del Tutorial de MATLAB) . . . | 8 |
| 2.4. Red Neuronal Artificial (Tomada del Tutorial de MATLAB) | 9 |
| 2.5. Arquitectura del MLP | 10 |
| 3.1. Ubicación estaciones ANGV y CR2R (cortesía de INGEOMINAS) | 13 |
| 3.2. Ejemplos de registros no incluidos | 14 |
| 3.3. Ejemplos de registros incluidos | 15 |
| 3.4. Estracción de ventana | 17 |
| 3.5. Izquierda. Segmento no ventaneado. Derecha Segmento ventaneado | 17 |
| 3.6. Gráficas de información acerca del proceso de entrenamiento | 22 |
| 3.7. Regreción lineal | 22 |
| 4.1. Regresiones para el lote de prueba 1.1 | 26 |
| 4.2. Regresiones para el lote de prueba 1.2 | 27 |
| 4.3. Regresiones para el lote de prueba 1.3 | 28 |
| 4.4. Regresiones para el lote de prueba 2.1 | 31 |
| 4.5. Regresiones para el lote de prueba 2.2 | 32 |
| 4.6. Regresiones para el lote de prueba 2.3 | 33 |
| 4.7. Regresiones para el lote de prueba 3.1 | 35 |
| 4.8. Regresiones para el lote de prueba 3.2 | 37 |
| 4.9. Regresiones para el lote de prueba 3.3 | 38 |

Introducción

Los registros sísmicos que se obtienen en los diferentes observatorios vulcanológicos del mundo, son en primera instancia clasificados en grupos sugeridos por la literatura especializada en el tema. Esta clasificación es a veces modificada de acuerdo con las particularidades de la actividad que sucede en cada volcán. Los criterios para la clasificación de los registros son generalmente establecidos partiendo de la representación de los mismos en el dominio de la frecuencia (espectro de amplitudes, espectrogramas) y en el dominio del tiempo (rubrica o firma de cada registro); también es importante el análisis comparativo de los registros de un mismo evento, captados en diferentes estaciones. La forma de clasificación en la gran mayoría de observatorios está sujeta al criterio de uno o varios expertos, quienes toman en cuenta las características de las representaciones anteriormente mencionadas. En este proceso frecuentemente surgen opiniones diferentes, esto le imprime cierto grado de subjetividad a la clasificación. Una forma de solucionar gran parte del problema sería construir un sistema experto que clasifique automáticamente los registros sísmicos, pero los parámetros para identificar los diferentes tipos tendrían que ser asignados por personas con experiencia en clasificación dejando nuevamente abierta la brecha de la subjetividad. En el presente trabajo la clasificación de los registros sísmicos se basa en los criterios utilizados en el Observatorio Vulcanológico de la ciudad de Pasto (OVSP) y se asume que los registros han sido correctamente clasificados en el pasado. Este trabajo se desarrolla para la discriminación de registros sísmicos asociados con eventos producidos por el volcán Galeras, entre los que se encuentran los registros correspondientes a eventos tipo LPS (asociados al movimiento de fluidos en el interior del volcán), VTA (relacionados con fractura de material en el edificio volcánico) y HBD (en los que se encuentra una fase asociada con movimiento de fluidos y otra con fractura de material cortical). Se pretende hacer un primer aporte al proceso que implica la implementación de un sistema experto en la clasificación de registros sísmicos asociados con la actividad del volcán Galeras. El desarrollo de este trabajo se llevó a cabo en tres fases bien diferenciadas. La primera consiste en la elección y extracción de los registros sísmicos de la base de datos de INGEOMINAS, de acuerdo con los requerimientos del tratamiento que se les aplicará. En la segunda etapa del trabajo se sigue con la preparación de los datos previo a su introducción al sistema que realizará la clasificación; en esta parte se reduce la dimensionalidad de cada registro con ayuda de la técnica de Coeficientes de Predicción Lineal (LPC) por sus siglas en inglés. La etapa final consiste en la implementación en MATLAB 6.0 de Redes Neuronales Artificiales (RNA) tipo Perceptrón Multicapa que después de un proceso de entrenamiento y prueba serán las encargadas de la discriminación entre los diferentes tipos

de registros. La justificación del uso de las RNA en este caso radica en que las características del problema se ajustan a las características generales que debe cumplir un problema para solucionarlo mediante el uso de RNA. Algunas de estas características son ¹:

- No se dispone de un conjunto de reglas sistemáticas que describan completamente el problema.
- Se dispone de muchos ejemplos o casos históricos.
- Los datos procedentes del problema son imprecisos o incluyen ruido.
- El problema es de elevada dimensionalidad.
- Si las condiciones de trabajo son cambiantes se puede hacer uso de la capacidad de las RNA para adaptarse a esos cambios, re-entrenando el sistema con nuevos ejemplos.

Es fácil asociar estas características al problema que se trata de resolver con el presente trabajo. Los registros seleccionados están asociados a eventos captados por los sismómetros de las estaciones Anganoy (ANGV) y Cráter (CR2R). Estos sismómetros son de Corto Periodo (CP). La discriminación entre registros tipo LPS y VTA se hace para eventos captados por las dos estaciones, poniendo en evidencia una característica importante de las RNA que es su capacidad adaptativa. La discriminación entre registros tipo LPS y HBD se hace para la estación ANGV unicamente.

Para futuros trabajos se sugiere extender la clasificación entre más de dos tipos de registros sísmicos, asociados o no con el fenómeno volcánico, tales como los registros tipo Tremor Volcánico, Tornillos, Tectónicos Locales, Tectónicos Regionales y Telesismos. Una vez entrenadas y probadas, las RNA se pueden implementar en-línea para la realización automática de la clasificación.

¹DEL BRIO, Bonifacio. SANZ, Alfredo. Redes Neuronales y Sistemas Difusos; Universidad de Zaragoza. 2002.

OBJETIVOS

0.1. Objetivos Generales

Utilizando una red neuronal tipo Perceptron Multicapa, discriminar, para los registros sísmicos de las estaciones de corto período Anganoy y Crater-2, entre eventos sísmicos tipo LPS ,VTA Y HBD que tienen como fuente el Volcán Galeras.

0.2. Objetivos Específicos

- Reducir la dimensionalidad de los registros sísmicos representándolos con ayuda de la técnica de Coeficientes de Predicción Lineal (LPC).
- Determinar la arquitectura de un Perceptrón Multicapa para la clasificación de registros sísmicos de la estación Anganoy.
- Determinar la arquitectura de un Perceptrón Multicapa para la clasificación de registros sísmicos de la estación Cráter-2.

ESTADO DEL ARTE

Con el advenimiento del Perceptrón Multicapa (1986) y la invención del algoritmo de entrenamiento denominado retro-propagación del error, se ha venido incrementando la cantidad de problemas de clasificación, en diferentes áreas, solucionados con ayuda de las RNA. El trabajo de Benbrahim, Daoudi y Benjelloun¹, está orientado a la discriminación entre registros tipos tectónico-local y explosión química, asociados con eventos ocurridos en Morocco, Estados Unidos. En su trabajo hacen uso de un Perceptrón Multicapa con configuración 20-7-2, probando con varias formas de representación de las señales de las cuales la que arroja mejores resultados es la representación denominada Escalograma de Ondicula de Sombrero Mejicano Modificado. Los resultados que obtienen son de un 88.75 % de aciertos en la clasificación. El trabajo de Del Pezzo, Esposito, Giudicepietro, Marinaro, Martín y Scarapetta², en Italia, es similar al citado anteriormente, pero en este caso se clasifica entre eventos tipo VTA originados en el volcán Vesuvio y explosiones producidas decenas de metros por debajo del nivel del mar por pescadores de la región. En dicho trabajo se hace uso de la técnica LPC para la representación de las señales y se diseña una arquitectura de red neuronal de tipo Perceptrón Multicapa con una configuración 105-6-1. El porcentaje de aciertos que logran es de un 92 %, promediado entre varios lotes de prueba. A nivel de Colombia el trabajo de Orozco, García, Duin y Castellanos³, persigue igual fin pero utilizando técnicas de representación de disimilitud sobre las cuales construyen clasificadores bayesianos para la discriminación de registros sísmicos producidos por el Volcán Nevado del Ruiz. En ese trabajo se discrimina en una primera etapa entre eventos tipo LPS y VTA y posteriormente entre eventos tipo LPS, VTA e IC (Avalanchas de Hielo). En la primera etapa obtuvieron aciertos entre un 85 % y 87 %, mientras que en la segunda los aciertos fueron entre un 81 % y 84 %; en ambos casos los resultados fueron obtenidos utilizando varios lotes de prueba.

¹BENBRAHIM, Mohammed. DAOUDI, Adil. BENJELLOUN, Khalid. IBENBRAHIM, Aomar. Discrimination of seismic signals using artificial neuronal networks; Enformatika v4 2005 ISSN 1305-5313; Morocco, Unite Estates.

²DEL PEZZO, Edoardo. ESPOSITO, Anna. GIUDICE, Flora. MARINARO, Maria. MARTINI, Marcello. SCARPETTA, Silvia. Discrimination of earthquakes and underwater explosions using neuronal networks.

³Orozco, Garía, Duin y Castellanos

Capítulo 1

COEFICIENTES DE PREDICCIÓN LINEAL (LPC)

¹Los LPC se obtienen mediante el análisis de predicción lineal basado en bloques (de muestras) de entrada así: sea $S(n)$ la señal de salida de un sistema cuya entrada es la señal de excitación $u(n)$. La muestra $S(n)$ es modelada como una combinación lineal de las salidas pasadas y de las entradas presente y pasadas. Esto puede ser expresado con ayuda de la siguiente ecuación de diferencias:

$$S(n) = \sum_{k=1}^p a_k S(n-k) + G \sum_{l=0}^q b_l u(n-l) \quad (1.1)$$

con $b_0 = 1$, donde la ganancia del filtro G y los coeficientes del filtro $\{a_k\}$, $\{b_l\}$ son los parámetros del sistema. El número p indica el número de muestras de salidas pasadas que se han considerado y también es el orden de la predicción lineal. La función de transferencia $H(z)$ se obtiene aplicando la transformada z a la ecuación (1):

$$H(z) = \frac{S(z)}{u(z)} = \frac{1 + \sum_{l=1}^q b_l z^{-l}}{1 - \sum_{k=1}^p a_k z^{-k}} \quad (1.2)$$

siendo $H(z)$ un modelo polo-cero.

Existen dos casos especiales en este modelo:

1. Si $b_l = 0$ y para $1 \leq l \leq q$, la ecuación (2) se reduce a un modelo todo-polo que se conoce como modelo autoregresivo.
2. Cuando $a_k = 0$ y para $1 \leq k \leq p$, $H(z)$ se reduce a un modelo todo-cero o modelo de promedio móvil.

La función de transferencia de un modelo todo-polo es:

$$H(z) = \frac{1}{1 - \sum_{k=1}^p a_k z^{-k}} \quad (1.3)$$

¹Este capítulo se basa en el artículo: ISLAM, Tamaña. Interpolation of Linear Prediction Coefficients for speech coding; McGill University. Montreal Canada. 2000.

Un modelo todo-polo es un buen estimador de un modelo polo-cero, lo que se argumenta de la siguiente manera: un sistema racional causal puede estar descrito por:

$$H(z) = G' H_{min}(z) H_{ap}(z) \quad (1.4)$$

donde G' es el factor de ganancia, $H_{min}(z)$ es la función de transferencia de un filtro de mínima fase y $H_{ap}(z)$ es la función de transferencia de un filtro todo-polo. Ahora la componente de mínima fase puede ser expresada como un sistema todo-polo:

$$H(z) = \frac{1}{1 - \sum_{i=1}^I a_i z^{-i}} \quad (1.5)$$

donde I es teóricamente infinito, pero generalmente puede tomar el valor de un entero pequeño que lo represente. La componente todo-paso contribuye únicamente a la fase, por consiguiente el modelo polo-cero puede ser estimado mediante un modelo todo-polo.

La transformada z inversa de (3) está dada por:

$$S(n) = \sum_{k=1}^p a_k S(n-k) + Gu(n) \quad (1.6)$$

si $G = 1$ la forma de la función de transferencia (3) es:

$$H(z) = \frac{1}{1 - \sum_{k=1}^p a_k z^{-k}} \quad (1.7)$$

$$H(z) = \frac{1}{A(z)} \quad (1.8)$$

donde

$$A(z) = 1 - \sum_{k=1}^p a_k z^{-k} \quad (1.9)$$

Los coeficientes del filtro $\{a_k\}$ se denominan coeficientes de predicción lineal (LPC). La señal de error $e(n)$ es la diferencia entre la señal de entrada y la señal estimada, por tanto:

$$e(n) = s(n) - \sum_{k=1}^p a_k s(n-k) \quad (1.10)$$

que en el dominio z equivale a:

$$E(z) = S(z)A(z) \quad (1.11)$$

1.1. Estimación de los LPC

El método más utilizado para la estimación de los LPC es el de autocorrelación, en el cual los primeros coeficientes $\{a_k\}$ del filtro se escogen de tal manera que la energía residual

se minimiza. Para este propósito se utiliza la técnica clásica de los mínimos cuadrados. La señal en el dominio del tiempo $S(n)$ se divide en segmentos sucesivos, que son sometidos a un proceso de ventaneo, esto es, cada segmento es multiplicado por una función $w(n)$ de longitud fija. La ventana más simple es la ventana rectangular que se define como:

$$w(n) = \begin{cases} 1, & 0 \leq n \leq N_w - 1, \\ 0, & \text{para los demás.} \end{cases}$$

Este tipo de ventana causa cambios abruptos en el dominio del tiempo y muestra irregularidades en el dominio de la frecuencia. La ventana de Hamming es bastante usada y arroja buenos resultados, esta se define como:

$$w(n) = \begin{cases} 0,54 - 0,46\cos\left(\frac{2\pi n}{N_w - 1}\right), & 0 \leq n \leq N_w - 1, \\ 0, & \text{para los demás.} \end{cases}$$

1.2. Método de autocorrelación

En primer lugar se toma un segmento de la señal que ha sido sometido al proceso de ventaneo, esto es:

$$S_w(n) = w(n)s(n) \quad (1.12)$$

Luego se minimiza la energía de la señal residual, la cual se define como:

$$E = \sum_{n=-\infty}^{\infty} e^2(n) \quad (1.13)$$

$$E = \sum_{n=-\infty}^{\infty} \left(S_w(n) - \sum_{k=1}^p a_k S_w(n-k) \right)^2 \quad (1.14)$$

Los valores de $\{a_k\}$ que minimizan E se encuentran igualando a cero las derivadas parciales de E respecto a $\{a_k\}$. Así:

$$\frac{\partial E}{\partial a_k} = 0 \quad (1.15)$$

para $k = 1, \dots, p$, entonces tenemos p ecuaciones con p variables desconocidas $\{a_k\}$ como muestra la siguiente ecuación:

$$\sum_{k=1}^p a_k \sum_{n=-\infty}^{\infty} S_w(n-i)S_w(n-k) = \sum_{n=-\infty}^{\infty} S_w(n-i)S_w(n) \quad (1.16)$$

con $1 \leq i \leq p$

Las ecuaciones lineales pueden ser expresadas en términos de la función de autocorrelación, ya que la función de autocorrelación del segmento ventaneado $S_w(n)$ esta dada por:

$$R(i) = \sum_{n=i}^{N_w-1} S_w(n)S_w(n-i) \quad (1.17)$$

con $1 \leq i \leq p$ donde N_w es la longitud de la ventana. Además, la función de autocorrelación es par:

$$R(i) = R(-i) \quad (1.18)$$

sustituyendo (19) en (18) tenemos:

$$R(i) = \sum_{k=i}^p R(|i-k|)a_k \quad (1.19)$$

con $1 \leq i \leq p$.

El conjunto de ecuaciones lineales (19) puede ser representado en forma matricial de la siguiente manera:

$$\begin{pmatrix} R(0) & R(1) & \cdots & R(p-1) \\ R(1) & R(0) & \cdots & R(p-2) \\ \vdots & \vdots & \cdots & \vdots \\ R(p-1) & R(p-2) & \cdots & R(0) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{pmatrix} = \begin{pmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{pmatrix} \quad (1.20)$$

que también podemos escribir como:

$$Ra = r \quad (1.21)$$

La matriz resultante es una matriz de *Teoplitz* donde los elementos a lo largo de la diagonal son iguales, esto permite resolver el sistema de ecuaciones mediante el algoritmo de Levinson-Durbin.

1.3. Procedimiento de Levinson Durbing

El algoritmo de Levinson resuelve un sistema de ecuaciones de la forma $Ax = b$, donde A es una matriz simétrica positiva de *Teoplitz* y b es un vector arbitrario. Durbin elaboró un algoritmo más eficiente llamado algoritmo recursivo de Levinson-Durbin. Este algoritmo necesita una forma especial de b conformada por algunos elementos de A , lo cual se logra utilizando la ecuación de autocorrelación. Sea $a_k(m)$ el k -ésimo coeficiente para un marco particular durante la m -ésima iteración. Entonces el algoritmo de Levinson-Durbin resuelve el siguiente conjunto de ecuaciones ordenadas recursivamente ($m = 1, 2, \dots, p$):

$$k(m) = R(m) - \sum_{k=1}^{m-1} a_k(m-1)R(m-k) \quad (1.22)$$

$$a_m(m) = k(m) \quad (1.23)$$

$$a - k(m) = a_k(m-1) - k(m)a_{m-k}(m-1) \quad (1.24)$$

con $1 \leq k \leq m$.

$$E(m) = (1 - k(m)^2)E(m-1) \quad (1.25)$$

donde inicialmente

$$E(0) = R(0) = a(0) = 0 \quad (1.26)$$

Para cada iteración, los coeficientes $a_k(m)$ con $k = 1, 2, \dots, m$ describen el predictor lineal óptimo de orden m -ésimo y el mínimo error $E(m)$ se reduce en un factor de $1 - k(m)^2$. Puesto que el error cuadrático nunca es negativo y $|k(m)| \leq 1$, se garantiza que las raíces de $A(z)$ estarán siempre dentro del círculo unidad. Así la síntesis del filtro $H(z)$, donde $H(z) = 1/A(z)$, será estable y por consiguiente el método de autocorrelación garantiza la estabilidad del filtro.

Capítulo 2

REDES NEURONALES ARTIFICIALES (RNA)

2.1. La neurona artificial

¹Se denomina neurona artificial a un dispositivo simple de cálculo el cual para cada vector de entrada procedente del exterior o de otras neuronas proporciona una salida única.

Un modelo estandar de neurona artificial esta constituido por los siguientes elementos:

- Entradas $x_j(t)$ constituidas por el vector de entrada.
- Pesos sinápticos w_{ij} representan la intensidad de interacción entre la neurona pre-sináptica y la neurona postsináptica.
- Regla de propagación $\sigma(w_{ij}, x_j(t))$, proporciona el valor del potencial postsináptico $h_i(t) = \sigma(w_{ij}, x_j(t))$ de la neurona i en función de sus pesos y entradas.
- Función de activación $f_i(a_i(t-1), h_i(t))$ proporciona el estado de activación actual $a_i(t) = f_i(a_i(t-1), h_i(t))$ de la neurona i en función de su estado anterior $a_i(t-1)$ y de su potencial postsináptico actual.
- Función de salida $F_i(a_i(t))$ proporciona la salida actual $y_i(t) = F_i(a_i(t))$ de la neurona i en función de su estado de activación. La operación de la neurona i es:

$$y_i(t) = F_i(f_i[a_i(t-1), \sigma_i(w_{ij}, x_j(t))]) \quad (2.1)$$

Para determinar por completo la operación anterior falta definir la función de activación que puede ser lineal, tipo escalón, sigmoidea, gaussiana, sinusoidal, entre otras. En vista de que el objetivo del presente trabajo es la clasificación mediante una RNA con arquitectura tipo

¹Este capitulo está basado en: DEL BRIO, Bonifacio. SANZ, Alfredo. Redes Neuronales y Sistemas Difusos; Universidad de Zaragoza. 2002. y HAYKIN, Simon. Neuronal Networks; McMaster University. 1994.

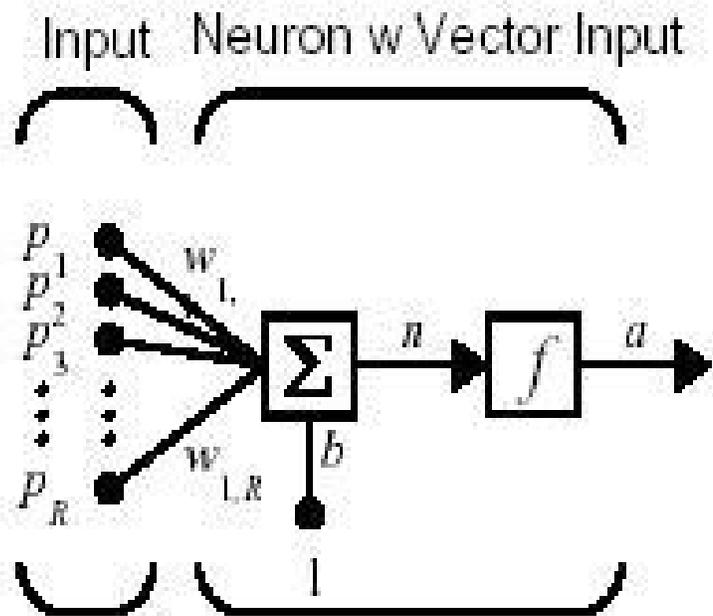


Figura 2.1: Modelo de Neurona Estandar (Tomada del Tutorial de MATLAB)

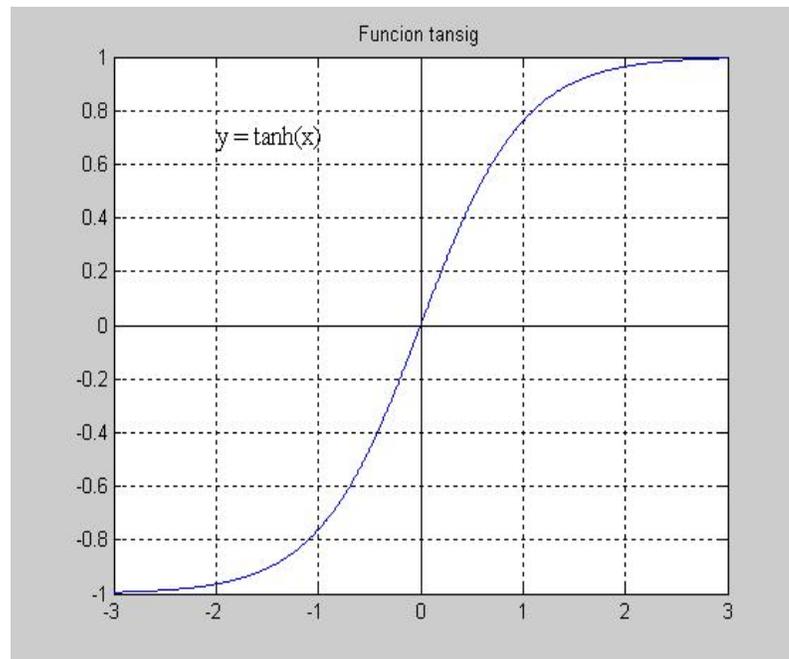


Figura 2.2: Gráficas de información acerca del proceso de entrenamiento

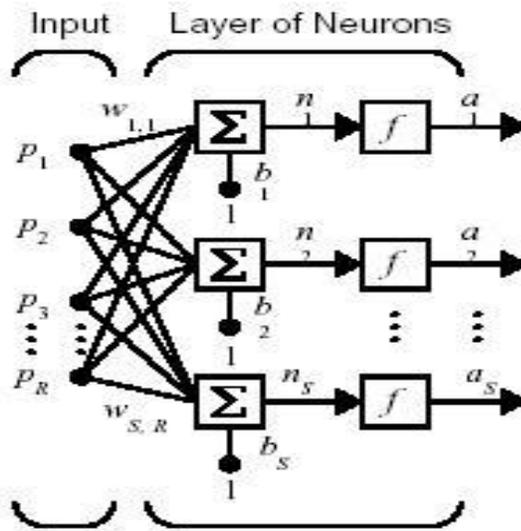


Figura 2.3: Capa de una Red Neuronal Artificial (Tomada del Tutorial de MATLAB)

Perceptrón Multicapa, la función de activación que se usa en la totalidad de las neuronas es de tipo sigmoideo que se define como sigue:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.2)$$

2.2. Redes Neuronales Artificiales

Se denomina arquitectura a la topología, estructura o patrón de conexionado de las neuronas. Estas se suelen agrupar en unidades estructurales denominadas capas. Las neuronas de una capa pueden agruparse, a su vez, formando grupos neuronales. Finalmente el conjunto de una o más capas constituye una red neuronal.

2.2.1. Definición de red neuronal

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

1. A cada nodo i se asocia una variable de estado x_i .
2. A cada conexión (i, j) de los nodos i y j se asocia un peso $w_{ij} \in \mathfrak{R}$.
3. A cada nodo i se asocia un umbral θ_i .
4. Para cada nodo i se define una función $f_i(x_i, w_{i,j}, \theta_i)$, 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.

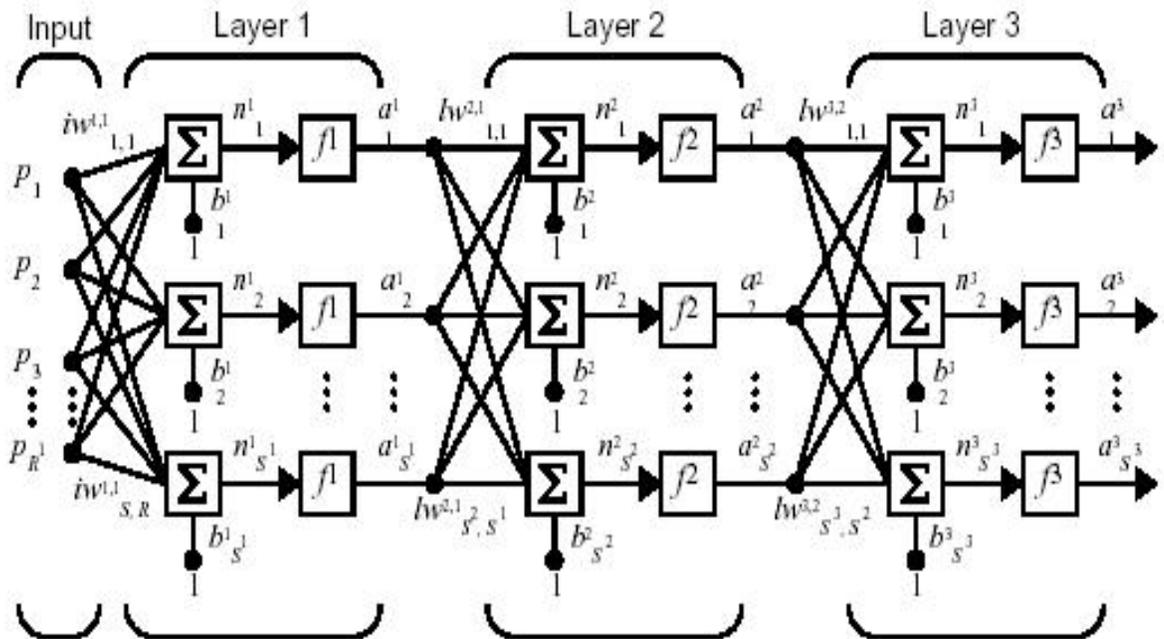


Figura 2.4: Red Neuronal Artificial (Tomada del Tutorial de MATLAB)

2.2.2. Modos de operación: recuerdo y aprendizaje

Se distinguen dos modos de operación en los sistemas neuronales: el modo de ejecución y el modo de aprendizaje o entrenamiento. Este último es de particular interés, pues una característica de las RNA es que se trata de sistemas entrenables, capaces de realizar un determinado tipo de procesamiento o cómputo aprendiéndolo a partir de un conjunto de patrones de aprendizaje o ejemplos.

2.2.3. Fase de aprendizaje

El aprendizaje es un proceso mediante el cual se produce el ajuste de los parámetros libres de la red. Durante el mismo la red es estimulada por su entorno. El entrenamiento o aprendizaje consiste en modificar los pesos sinápticos siguiendo una cierta regla de aprendizaje, construida normalmente a partir de la optimización de una función de error, que mide la eficacia de la operación de la red. Si denominamos w_{ij} al peso que conecta la neurona presináptica j con la neurona postsináptica i en la iteración t , el algoritmo de aprendizaje, en función de las señales que en el instante t llegan del entorno, proporcionará el valor $\Delta w_{ij}(t)$ que da la modificación que se debe incorporar en dicho peso, el cual quedará actualizado en la forma:

$$\Delta w_{ij}(t+1) = w_{ij}(t) + \Delta w_{ij}(t) \quad (2.3)$$

El proceso de aprendizaje es usualmente iterativo. Para este trabajo se hace uso del modo de aprendizaje llamado Aprendizaje Supervisado que se describe así: Sea $E[W]$ un funcional

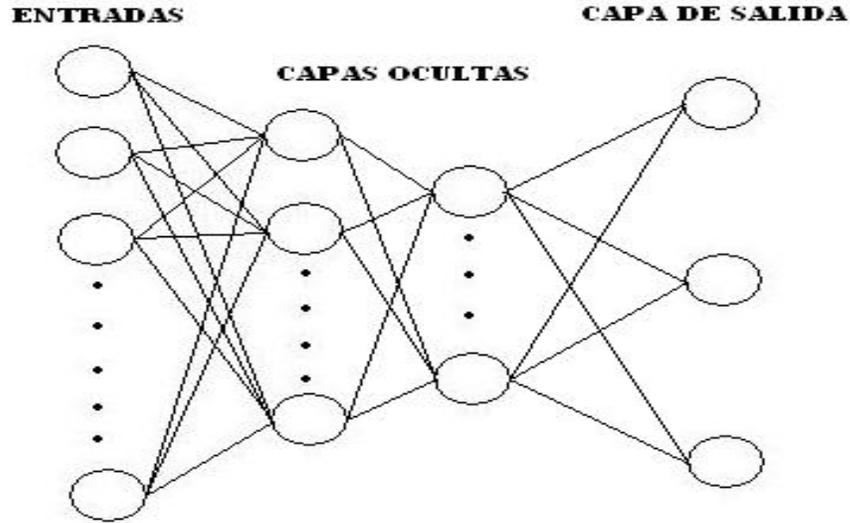


Figura 2.5: Arquitectura del MLP

que representa el error esperado en la operación de la red, expresado en función de sus pesos sinápticos W . Se pretende estimar una función multivariable desconocida $f : \mathfrak{R}^n \rightarrow \mathfrak{R}^m$ que representa a la red a partir de muestras $(x, y) (x \in \mathfrak{R}^n, y \in \mathfrak{R}^m)$ tomadas aleatoriamente. Para esto se realiza la minimización iterativa de $E[W]$ mediante una aproximación estocástica.

2.2.4. Fase de recuerdo o ejecución: estabilidad.

Una vez que el sistema ha sido entrenado el aprendizaje se desconecta, por lo que los pesos y estructuras quedan fijos. En las redes unidireccionales, ante un patrón de entrada, las neuronas responden proporcionando directamente la salida del sistema.

2.2.5. El Perceptrón Multicapa (MLP)

Este tipo de red se ubica en el grupo de redes unidireccionales, no recurrentes. Se la conoce por sus buenos resultados en problemas de aproximación de funciones y problemas de clasificación. La arquitectura del MLP consta de una capa de entrada, una o más capas ocultas y una capa de salida. La función de transferencia que se usa en las neuronas del MLP son generalmente de tipo sigmóideo. La estructura del MLP se presenta en las figuras (10) y (11). Denominaremos x_i a las entradas de la red, y_i a las salidas de la capa oculta z_k a las de la capa final; t_k serán las salidas objetivo. Por otro lado, w_{ij} son los pesos de la capa oculta y θ_j sus umbrales, w'_{kj} los pesos de la capa de salida y θ'_k sus umbrales. La operación de un MLP con una capa oculta y neuronas de salida lineal es:

$$z_i = \sum_j w'_{kj} y_j - \theta'_i = \sum_j w'_{kj} f\left(\sum_i w_{ji} x_i - \theta_j\right) - \theta'_i \quad (2.4)$$

Siendo $f(\cdot)$ de tipo sigmoideo. Esta arquitectura suele entrenarse mediante el algoritmo denominado retropropagación de errores (BP)² o bien haciendo uso de algunas de sus variantes.

²Una descripción completa de este algoritmo puede encontrarse en HAYKIN, Simon. *Neuronal Networks*; McMaster University. 1994.

Capítulo 3

METODOLOGIA

En el presente trabajo, la clasificación de los diferentes tipos de registros sísmicos se ha desarrollado en tres etapas. En la primera se hace la extracción y elección de las trazas desde la base de datos del OVSP, las cuales corresponden a sismómetros de corto periodo (CP) y se encuentran almacenadas en la base del Observatorio en un formato conocido como WVG (Waveform). Debido a que MATLAB no acepta este formato, los archivos han sido convertidos a un formato compatible con MATLAB. En la segunda etapa se reduce la dimensionalidad de cada una de las trazas. Por último, se implementan diferentes arquitecturas de RNA para lograr los mejores resultados posibles en la clasificación. A continuación se describe de forma detallada cada una de las etapas enunciadas anteriormente.

3.1. DATOS

El volcán Galeras es monitoreado permanentemente por el OVSP. Entre las tareas de monitoreo está la de recolección de registros sismológicos. Para este fin el Observatorio cuenta con una red de estaciones en el entorno del edificio volcánico que se encarga de registrar los eventos sísmicos que pueden o no ser originados en el volcán Galeras, pero que afectan la región donde se encuentra instalada la red.

Cada estación sismológica cuenta con un sismómetro que puede ser de banda ancha (BA) o de corto período (CP). Entre estas estaciones se encuentran las estaciones de Anganoy (ANGV) y Cráter (CR2R), ambas dotadas con sismómetros CP. Son los registros de estas estaciones con los que se trata en el desarrollo de este trabajo. La importancia de estas estaciones radica en que ambas han sido estaciones base de clasificación en el Observatorio, desempeñando este papel en la actualidad la estación ANGV. Los registros son transmitidos por medios telemétricos desde las estaciones hasta las instalaciones del OVSP, donde se realiza un muestreo de las señales. La frecuencia de muestreo es de 100 Hz. La señal muestreada se guarda en segmentos de una duración aproximada de 120 s, por lo que cada archivo, en formato WVG, cuenta con 12000 muestras aproximadamente.

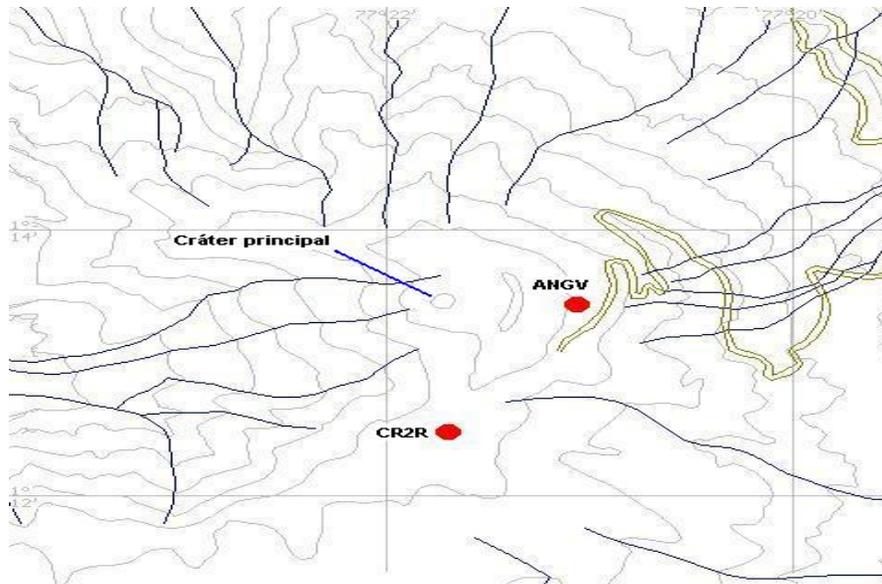


Figura 3.1: Ubicación estaciones ANGV y CR2R (cortesía de INGEOMINAS)

3.1.1. Elección y extracción de las trazas

En este trabajo se han tomado registros correspondientes al intervalo temporal comprendido entre 01/06/05 y 30/07/06. Estos registros servirán para entrenar y probar las RNA. La elección de este intervalo temporal es debido a que utilizando registros de entrenamiento y prueba próximos al presente, se espera que la clasificación de nuevos registros arroje mejores resultados ya que es probable que las características de los registros más antiguos sean diferentes de las de los registros recientes. Los criterios para la elección de las trazas que formarán parte de los conjuntos de entrenamiento y prueba son:

- Los registros no debe contener anomalías tales como picos asociados a pulsos de voltaje no deseados producidos por el equipo, o las señales de calibración utilizadas en el control del correcto funcionamiento del equipo.
- Los registros no deben rebasar el nivel de saturación.
- Cada archivo debe contener un solo evento de cualquiera de los tres tipos LPS, VTA o HBD.
- El registro del evento sísmico debe tener una duración mínima de 22.5 s después de su amplitud máxima

Los archivos que contienen los registros se convirtieron de formato WVG a formato txt, con ayuda del programa Sudasc que INGEOMINAS utiliza para tal fin. Luego, en MATLAB se creó una rutina llamada estrac que extrae de la hoja de texto únicamente los registros correspondientes a las estaciones de ANGV y CR2R. A cada vector de datos se aplica la rutina event3 creada para informar si el evento tiene los 22.5 s después de la máxima amplitud. Estas trazas se visualizan en una gráfica para detectar la presencia de picos de voltaje no deseados

inducidos por el equipo y que no corresponden al evento. De esta manera se analizan uno por uno los registros de las estaciones ANGV y CR2R. La teoría de RNA sugiere que el número de ejemplos de entrenamiento debe ser el mayor posible, por tanto, se ha decidido que 500 registros para cada tipo de evento es un número suficiente si lo comparamos con trabajos similares en los que se utilizaron alrededor de 70 registros de cada tipo. En total se utilizaron 1500 registros de la estación de ANGV y 1000 de la estación de CR2R, ya que para esta última no se trabajaron los registros de eventos tipo HBD. Es importante señalar que cada evento es registrado por ambas estaciones y se trabajaron solamente eventos registrados por las dos estaciones. En las figuras 6 y 7 se presentan ejemplos de registros que no fueron incluidos y de los que fueron incluidos en el los conjuntos de entrenamiento y prueba.

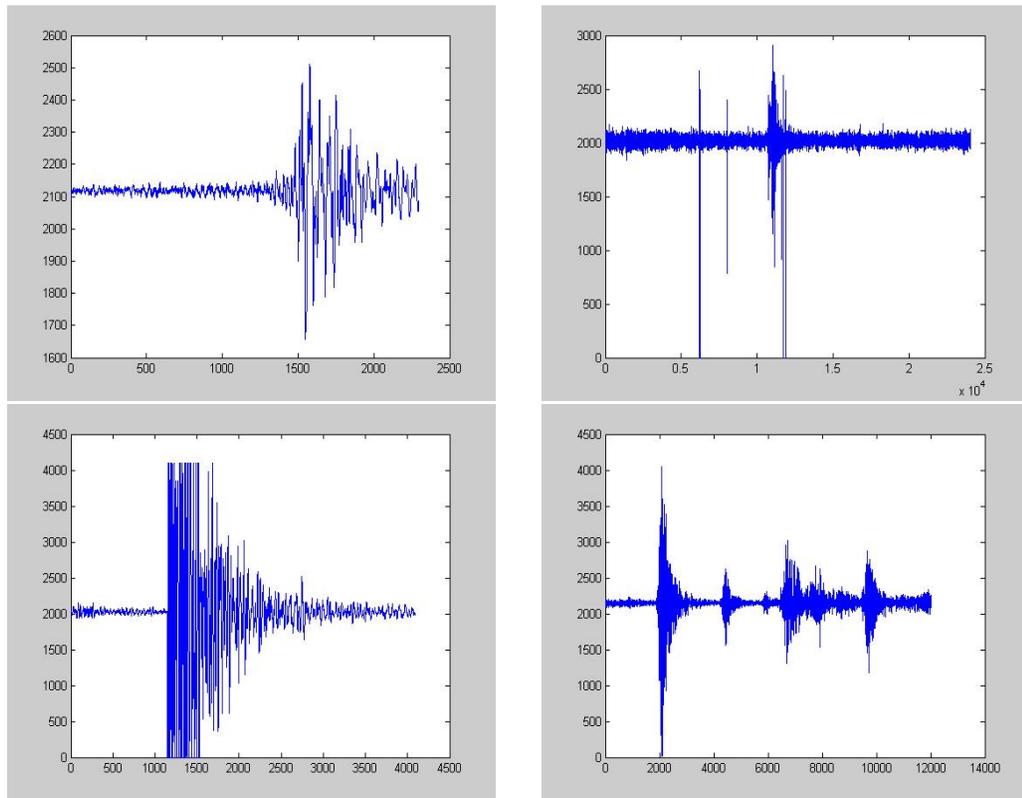


Figura 3.2: Ejemplos de registros no incluidos

Una vez elegidos los registros que formarán los conjuntos de entrenamiento se procede a dividirlos en lotes de 500 registros del mismo tipo (LPS, VTA o HBD) para la extracción de los LPC de cada evento.

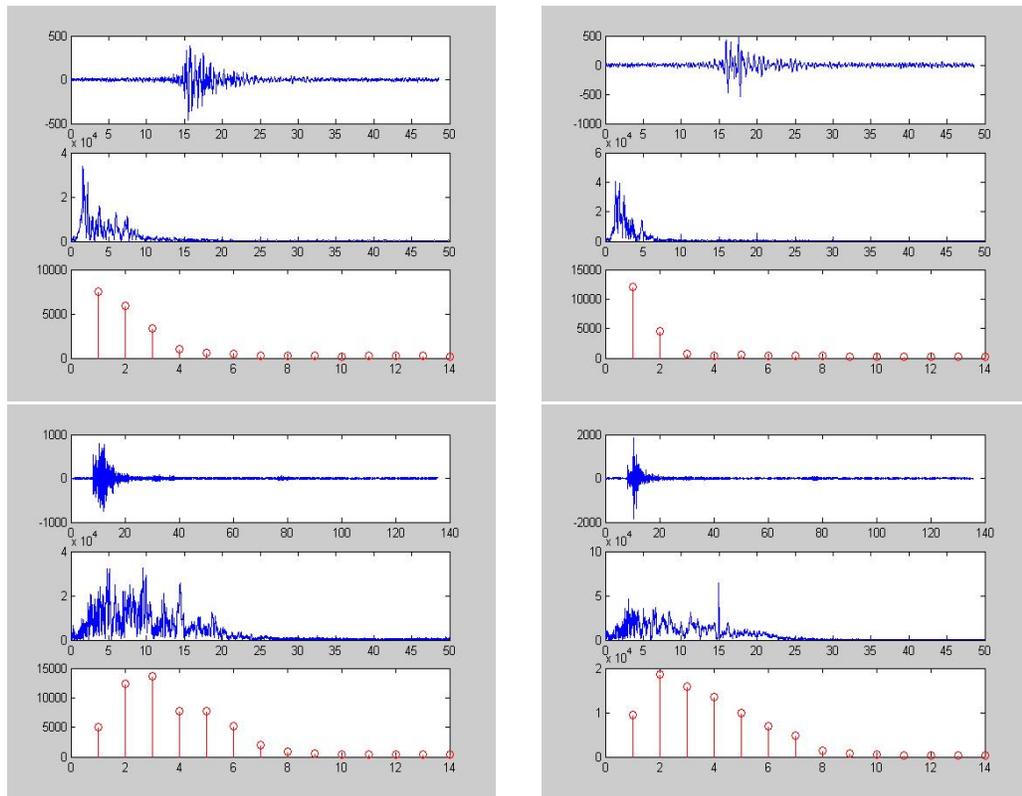


Figura 3.3: Ejemplos de registros incluidos

3.2. REPRESENTACIÓN Y REDUCCIÓN DE DIMENSIONALIDAD

El vector de cada traza, como se dijo anteriormente, tiene 1200 elementos aproximadamente. Lo que sigue en el procedimiento es reducir la dimensión de este vector. Para conseguir esto se crearon las rutinas `autoco`, `autocolp`, `autocovt` y `autocohb`, que se encargan del recorte y representación de la señal mediante los LPC. Las dos últimas letras del nombre de cada rutina indican a que tipo de señal serán aplicadas: `autocolp` se aplica a los registros tipo LPS, `autocovt` a los registros tipo VTA y `autocohb` a los registros tipo HBD. Cada rutina admite lotes de hasta 500 registros de un mismo tipo. Los lotes se encuentran ubicados en la carpeta `work` de MATLAB y desde allí las rutinas `autoco` los toma para procesarlos.

El procesamiento que las rutinas `autoco` aplican a los registros seleccionados es el siguiente:

- Extrae desde el archivo de texto las columnas correspondientes a las trazas de ANGV y CR2R ubicándolas en el workspace de MATLAB. Les asigna un nombre con las siguientes características: La primera letra (C) indica que se trata de un vector que contienen los LPC para ese registro. La segunda letra L, V o H indica el tipo de evento, las dos letras siguientes AN o CR indican la estación a la que pertenecen, finalmente

cada vector tiene un número de tres cifras para enumerar los registros correspondientes a un mismo tipo de evento y a una misma estación. Los vectores están ordenados de acuerdo a la fecha del evento asociado.

- Encuentra la amplitud máxima del registro y toma desde 5 segundos antes hasta 22.5 segundos después de la misma. Por tanto, se cuenta con una señal de 27.5 s de duración, lo que significa que un vector contiene 2750 muestras.
- Divide el vector de 27.5 s en 11 vectores de 2.5 s (250 muestras) y aplica una ventana de Hamming a cada uno de los 11 vectores. Mediante la rutina LPC de MATLAB se calculan 9 LPC para cada vector de 2.5 s y se forma un vector de 99 elementos que representan la señal.

De esta manera se concluye la representación de los registros con ayuda de los LPC. Ahora se cuenta con 2500 vectores que representan los registros inicialmente seleccionados de las estaciones ANGV y CR2R. Estos vectores son los que conformaran los conjuntos de entrenamiento y prueba en la siguiente etapa, donde se buscan las mejores arquitecturas de red para garantizar una buena discriminación.

Es importante justificar dos puntos delicados que se han visto involucrados en los procedimientos anteriores.

3.2.1. Elección del tamaño de los intervalos de registro.

Se bien es cierto que el tiempo de duración de los diferentes eventos sísmicos no es el igual, se hace necesario estimar un tiempo de duración característico igual para todos los eventos ya que una RNA tipo Perceptrón Multicapa requiere que los vectores de entrada sean de igual tamaño. Se ha elegido una ventana de 27.5 s ya que la mayoría de los eventos analizados se encuentran en este rango de duración. Se han descartado los eventos registrados al final de cada archivo que no alcanzan 22.5 s de duración desde su máxima amplitud. De acuerdo con los criterios utilizados en el OVSP, los eventos tipos Tectónico-Local y VTA originados por fracturas del material cortical, se distinguen por la diferencia de los tiempos de arribo de las ondas S y P. Esta diferencia se denota como S-P. Cuando S-P es mayor que 5 s el evento es considerado Tectónico-Local, si es menor de 5 s el evento se considera VTA. Por lo general, el arribo de la onda S aproximadamente coincide con la máxima amplitud del evento, por tanto, se ha asumido con fines prácticos que los eventos tipo VTA comienzan 5 s antes de la amplitud máxima. En la figura 8 se muestra un ejemplo del intervalo de registro elegido.

3.2.2. Elección del número de LPC para cada segmento

La completa representación de la señal mediante los LPC se logra con un número infinito de ellos. Sin embargo, una aproximación aceptable se consigue limitando el número de los LPC. En la figura 9 se ilustra la dependencia del error respecto al número de los LPC para un

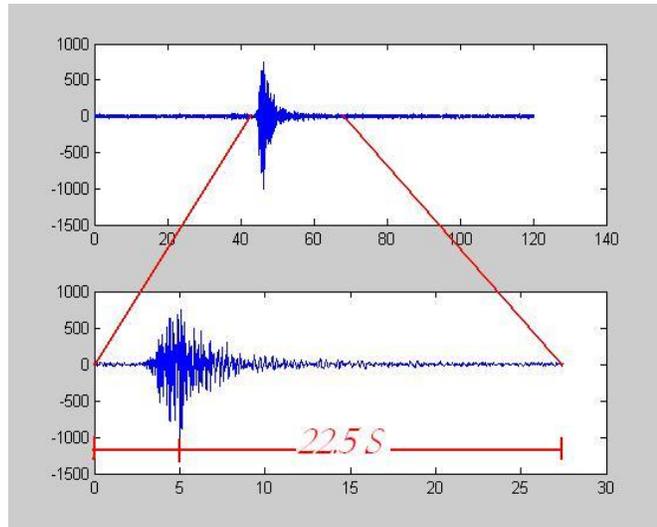


Figura 3.4: Estracción de ventana

evento dado. Se observa un comportamiento asintótico horizontal del error cuando el número de coeficientes es aproximadamente mayor que 9. Una mayor calidad de la representación de los segmentos se lograría con un número elevado de LPC, pero con el fin de reducir el tiempo de cómputo en la fase de entrenamiento del Perceptón Multicapa, la dimensión de los vectores de entrada debe ser moderada. Teniendo en cuenta estos dos factores se ha optado por representar cada segmento con 9 LPC, excluyendo en todos los 11 segmentos el primer coeficiente que es igual a 1 ya que el Perceptón Multicapa no los utiliza para distinguir características entre los vectores.

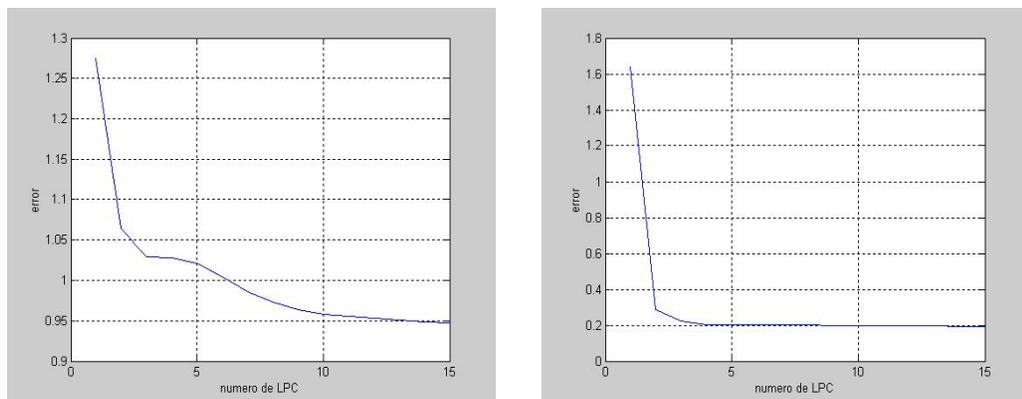


Figura 3.5: Izquierda. Segmento no ventaneado. Derecha Segmento ventaneado

3.3. CLASIFICACION

Para construir un clasificador con ayuda de un Perceptrón Multicapa las neuronas que lo componen deben ser de tipo sigmoideo. La capa de entrada tiene 99 elementos que corresponden a la dimensión de los vectores que representan los registros. Para determinar la mejor arquitectura del clasificador se requiere escoger adecuadamente el número de capas ocultas y de neuronas en dichas capas. Para la clasificación entre 2 tipos de registros, la capa de salida consta de 1 neurona. El algoritmo utilizado en los entrenamientos es una modificación del algoritmo de retro-propagación denominado Regularización, sugerido por el tutorial de MATLAB.

Se establecen arquitecturas de redes neuronales para las siguientes clasificaciones:

1. Clasificación entre eventos tipo LPS y VTA para la estación ANGV.
2. Clasificación entre eventos tipo LPS y VTA para la estación CR2R.
3. Clasificación entre eventos tipo LPS y HBD para la estación ANGV.

El establecimiento de la arquitectura de una RNA requiere de un procedimiento de entrenamiento y prueba. En cada entrenamiento se utilizaron 900 vectores: 450 de un tipo de evento y 450 del otro. Un entrenamiento sirve para establecer los valores finales de los pesos en la sinapsis a partir de unos valores aleatorios iniciales, diferentes para cada entrenamiento. Con un mismo lote de vectores se realizan 5 entrenamientos, obteniéndose 5 configuraciones de pesos finales diferentes. Esto último muestra que los pesos finales dependen en cierta medida de los pesos aleatorios iniciales.

Una vez finalizada la etapa de entrenamiento, las RNA se someten a prueba utilizando un mismo lote de 100 vectores de entrada desconocidos: 50 de un tipo de evento y 50 del otro. La cantidad de aciertos mide la efectividad de las RNA en su función de clasificación. Se escoge la RNA cuya configuración de pesos conlleve al mejor resultado.

El procedimiento de entrenamiento y prueba descrito anteriormente se llevó a cabo 3 veces con lotes de entrenamiento y prueba diferentes. Por tanto, para cada caso de clasificación, se aplicaron un total de 15 entrenamientos con sus respectivas pruebas y se escogieron tres RNA. El porcentaje promedio de aciertos es el promedio aritmético de los porcentajes de acierto de las 3 redes seleccionadas.

La rutina del procedimiento de entrenamiento y prueba es la siguiente:

1. En el workspace del programa MATLAB se cargan los vectores con los LPC que representan los registros de los tipos de eventos a discriminar.
2. Los vectores de entrenamiento se ubican en una matriz, igualmente los vectores de prueba.

3. Todos los vectores son normalizados, esto es exigido por el algoritmo de entrenamiento que se utiliza.
4. Se construye el objeto de red (RNA) con una determinada arquitectura.
5. Se introduce el primer lote de entrenamiento al objeto neuronal.
6. Se procede a la simulación del Perceptrón entrenado, haciendo uso del lote de vectores prueba.
7. Se definen y calculan los parámetros que caracterizan la efectividad del procedimiento de entrenamiento.
8. Se inicializan los pesos sinápticos y se vuelve al paso 1.

A continuación se explica un ejemplo de la rutina de entrenamiento y prueba utilizando una arquitectura de red, un lote de entrenamiento y un lote de prueba. El ejemplo muestra el procedimiento necesario para construir un clasificador aplicable a la discriminación de registros tipo LPS y VTA para la estación CR2R. La lista ordenada de comandos que se ejecutan en MATLAB para el desarrollo del procedimiento es la siguiente:

```

>> clear

>> load matin

>> load tar

>> a=matCLV(:,1:500);

>> b=matCLV(:,601:1000);

>> pt=matCLV(:,501:600);

>> p=[a b];

>> t=TLV(:,1:900);

>> tt=TLV(:,901:1000);

>> [pn,pmin,pmax,tn,tmin,tmax]=premnmx(p,t);

>> ptn=tramnmx(pt,pmin,pmax);

>> net1=newff(minmax(pn), [3 1], {'tansig' 'tansig'}, 'trainbr');

```

```

>> net1.trainParam.epochs=200;

>> net1.trainParam.show=20;

>> net1=init(net1);

>> [net1, tr1]=train(net1, pn, tn);

TRAINBR, Epoch 0/200, SSE 1271.43/0, SSW 11.9799, Grad
5.76e+002/1.00e-010, # Par 3.04e+002/304
TRAINBR, Epoch 20/200, SSE 0.0191776/0, SSW 132.31, Grad
3.11e-001/1.00e-010, # Par 9.50e+001/304
TRAINBR, Epoch 37/200, SSE 1.30562e-012/0, SSW 415.592, Grad
5.55e-011/1.00e-010, # Par 1.04e+002/304

TRAINBR, Minimum gradient reached.

>> c=sim(net1, ptn);

>> [m, b, r]=postreg(c, tt)

m

= 0.9400

b

= -0.0600

r

= 0.9417

```

A continuación se describen los comandos de la rutina anterior.

El comando *clear* limpia el workspace.

La orden *load matn* carga en el workspace las matrices que contienen el conjunto de vectores de entrenamiento y prueba que se utilizan durante el desarrollo de la rutina. Los vectores se intercalan por tipo de evento en cada matriz.

La orden *load tar* carga en el workspace un conjunto de matrices que corresponden a los objetivos del aprendizaje. Para cada clasificación la matriz consta de un vector fila con 1000 elementos iguales a los números 1 y -1 intercalados. En este ejemplo los vectores que re-

presentan a los registros tipo LPS tienen como objetivo el elemento 1 y los vectores que representan los registros VTA el elemento -1. De esta manera cuando se proceda a la simulación de esta RNA con el lote de prueba, el resultado será un vector fila cuyos elementos pertenecen al intervalo $[-1, 1]$. La proximidad de estos elementos a los valores extremos de dicho intervalo indica el tipo de registro al que pertenece el vector de prueba.

La matriz p de dimensiones 99×900 está formada por los vectores asociados con los eventos tipos LPS y VTA intercalados entre sí. En las matrices a y b se han incluido los primeros 500 y los últimos 400 vectores de la matriz $matCLV$. Esta última guarda todos los vectores requeridos en este ejemplo de clasificación. La matriz p corresponde al lote de entrenamiento. La matriz de prueba pt se forma con los vectores de la matriz $matCLV$ que van desde la columna 501 hasta la 600 y por tanto su dimensión es de 99×100 .

El vector t contiene los objetivos de aprendizaje anteriormente mencionados. El vector tt contiene información acerca de la clasificación correcta para el lote de prueba. Este vector será utilizado en la valoración de la eficacia de la RNA. El comando `premmx` normaliza en el rango $[-1, 1]$ los vectores de entrenamiento, así como también el vector de objetivos. Como salida, esta orden da la matriz pn que contiene los 900 vectores de entrenamiento normalizados, también nos da los vectores $pmin$ y $pmax$ que muestran los valores mínimo y máximo que toman las filas de la matriz p . La orden `trammmx` normaliza los vectores de la matriz de prueba, para lo cual utiliza los vectores $pmin$ y $pmax$. La matriz ptn consta de los vectores de prueba normalizados.

La orden `newff` crea una red neuronal artificial unidireccional cuyos argumentos significan que cuenta con una capa oculta de 3 neuronas y una capa de salida con una neurona. Todas estas neuronas son de tipo sigmoideo (`tansig`) y durante el entrenamiento se aplica un algoritmo de retro-propagación modificado con regularización (`trainbr`). El argumento `minmax(pn)` de la orden `newff` toma los valores mínimo y máximo de cada uno de los vectores introducidos a la red. Los comandos `net1.trainParam.epochs` y `net1.trainParam.show` fijan respectivamente el número máximo de iteraciones durante el entrenamiento y el intervalo entre las iteraciones presentando información a cerca de la marcha del entrenamiento. La orden `init` inicializa los pesos sinápticos antes del inicio del proceso iterativo, asignándoles valores aleatorios. El comando `train` da comienzo al entrenamiento de la red que llamada `net1`. Nótese que los argumentos de esta función son `net1` (la red), pn (el lote de entrenamiento) y tn (el vector de objetivos). Los vectores de entrenamiento ingresan de uno en uno a la red y cuando todos han pasado por la red, esta calcula el valor de ajuste de los pesos con el fin de disminuir el error que la red comete al asociarlos con los objetivos. Este procedimiento se lleva a cabo iterativamente hasta que el gradiente de la curva de error alcance un valor mínimo. Con esto el proceso se detiene automáticamente. La nueva configuración de los pesos de las sinapsis es almacenada en `net1`. La salida `tr1` contiene información a cerca del entrenamiento. Al ejecutar el comando `train` se despliegan tres gráficas con información a cerca del entrenamiento.

En la primera gráfica de la figura 3.5 muestra la curva de la suma de los errores medios cuadráticos con respecto al número de iteraciones. La segunda curva muestra la suma de los cuadrados de los pesos sinápticos con respecto al número de iteraciones. La última gráfica

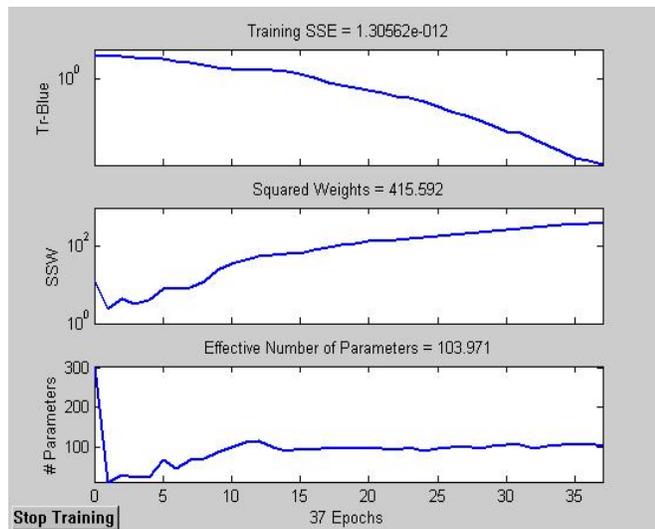


Figura 3.6: Gráficas de información acerca del proceso de entrenamiento

representa el número de parámetros de la red que fueron efectivamente utilizados durante el ajuste de los pesos. Esta información se muestra también en texto cada 20 iteraciones. Igualmente en el texto se indican el número de iteraciones necesarias para alcanzar el mínimo gradiente y el valor de este último.

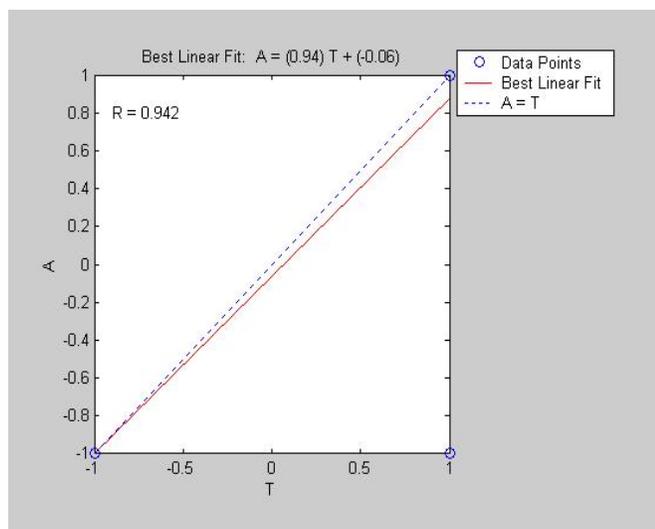


Figura 3.7: Regresión lineal

Concluido el entrenamiento, la red está lista para su aplicación en la clasificación de los vectores de prueba elegidos, acerca de los cuales la red no tiene ningún tipo de información (obviamente el entrenador conoce de antemano qué vectores en el lote de prueba están aso-

ciados con registros tipo LPS o VTA). La orden *sim* ejecuta la simulación del Perceptrón creado. Sus argumentos son *net1* (la red) y *ptn* (el lote de vectores de prueba). El vector *c* almacena los resultados de la clasificación.

En este ejemplo el Perceptrón cometió 3 equivocaciones en las posiciones 1, 33 y 79, donde clasificó eventos LPS como eventos VTA. Se puede concluir que con este Perceptrón se ha conseguido una generalización bastante buena ya que el porcentaje de aciertos es de un 97 %. Finalmente la orden *postreg* aplica una técnica de regresión lineal para encontrar la mejor recta que ajuste los valores encontrados por la red a los valores esperados. La figura 3.6 muestra la recta resultante de la regresión lineal. Las letras *m*, *b* y *r* representan respectivamente los valores de la pendiente, intercepto y coeficiente de correlación. Se observa que el coeficiente de correlación *r* tiene un valor de 0.9417 lo que también evidencia los buenos resultados obtenidos.

Al aplicar nuevamente el comando *init*, los pesos son ajustados aleatoriamente y se puede iniciar un nuevo entrenamiento.

Capítulo 4

RESULTADOS

La metodología del ejemplo anterior se aplicó a la solución de los diferentes problemas de clasificación. Los resultados obtenidos se listan en 15 tablas que se anexan en CD-ROM. Las tablas presentan la siguiente información:

- Columna 1. Nombres de los vectores de prueba utilizados en la simulación, en el orden en que fueron presentados a la red.
- Columna 2. Vector de los resultados esperados.
- Columna 3. Vector resultante de la simulación con la primera inicialización de la red.
- Columna 4. Errores cometidos por la red, los cuales se marcan con una x.

En las columnas de la 5 a la 12 se repite la información de las columnas 3 y 4 para cuatro inicializaciones diferentes. Además de esta información, se presentan los siguientes datos: mejor configuración de red, parámetros m , b , r y tiempo de entrenamiento asociado con cada inicialización.

4.1. Clasificación 1

En este problema se discrimina entre registros tipo LPS y VTA para la estación ANGV. La arquitectura de red 99-3-1 arrojó los mejores resultados.

Para presentar los resultados correspondientes a la clasificación 1 se hace las siguientes convenciones:

Lote de prueba 1.1. Desde el vector 1 hasta el 100 de la matriz que contiene el total de los vectores

Lote de prueba 1.2. Desde el vector 5011 hasta el 600 de la matriz que contiene el total de los vectores.

Lote de prueba 1.3. Desde el vector 901 hasta el 1000 de la matriz que contiene el total de los

vectores

Los resultados para cada inicialización y lote de prueba son:

| | Init 1 | Init 2 | Init 3 | Init 4 | Init 5 |
|----------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Aciertos | 98 % | 98 % | 98 % | 100 % | 98 % |
| m | 0.9655 | 0.9592 | 0.96 | 0.9982 | 0.96 |
| b | 5.50×10^{-3} | 7.95×10^{-4} | 2.62×10^{-5} | 1.80×10^{-3} | 1.03×10^{-5} |
| r | 0.9706 | 0.96 | 0.96 | 0.9998 | 0.96 |
| time | 14.92 | 16.06 | 15.29 | 14.5 | 15.26 |

Cuadro 4.1: Resultados lote de prueba 1.1

En la figura 4.1 se presentan las rectas obtenidas con la aplicación del método de regresión lineal ordenadas por inicializaciones (Init) de izquierda a derecha.

| | Init 1 | Init 2 | Init 3 | Init 4 | Init 5 |
|----------|-----------------------|------------------------|------------------------|------------------------|-----------------------|
| Aciertos | 95 % | 97 % | 96 % | 97 % | 96 % |
| m | 0.9002 | 0.9338 | 0.9217 | 0.9334 | 0.92 |
| b | 1.98×10^{-2} | -1.38×10^{-2} | -1.70×10^{-3} | -1.13×10^{-2} | 3.20×10^{-7} |
| r | 0.9006 | 0.9379 | 0.9231 | 0.9405 | 0.9206 |
| time | 16.62 | 18.67 | 13.02 | 15.78 | 15.77 |

Cuadro 4.2: Resultados lote de prueba 1.2

En la figura 4.2 se presentan las rectas obtenidas con la aplicación del método de regresión lineal ordenadas por inicializaciones (Init) de izquierda a derecha.

| | Init 1 | Init 2 | Init 3 | Init 4 | Init 5 |
|----------|------------------------|------------------------|------------------------|------------------------|------------------------|
| Aciertos | 93 % | 93 % | 94 % | 96 % | 95 % |
| m | 0.8602 | 0.8699 | 0.8784 | 0.9204 | 0.8733 |
| b | -2.00×10^{-2} | -3.00×10^{-2} | -3.84×10^{-2} | -2.20×10^{-2} | -3.33×10^{-2} |
| r | 0.8605 | 0.874 | 0.8813 | 0.9373 | 0.8787 |
| time | 13.25 | 14.4 | 11.58 | 12.17 | 12.95 |

Cuadro 4.3: Resultados lote de prueba 1.3

En la figura 4.3 se presentan las rectas obtenidas con la aplicación del método de regresión lineal ordenadas por inicializaciones (Init) de izquierda a derecha.

Las tablas 4.1.1, 4.1.2 y 4.1.3 son los resultados para los tres lotes de entrenamiento. Ahora para cada lote de prueba se escoge la mejor configuración de pesos. Para el lote del 1 al 100 esta configuración es la asociada a la inicialización número 4. Para el lote del 501 al 600 es

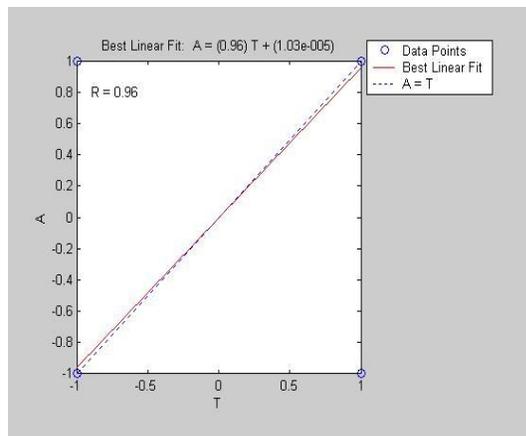
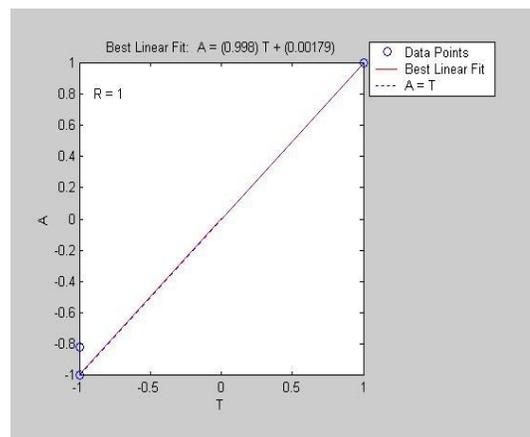
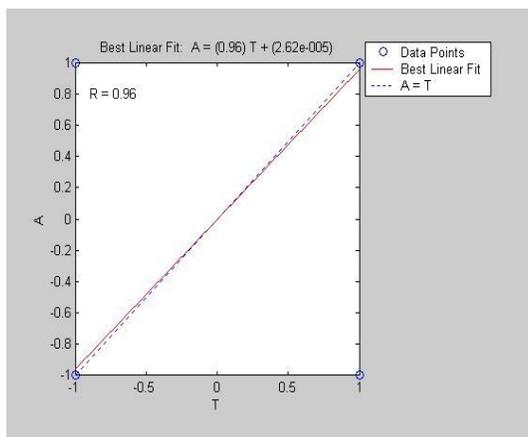
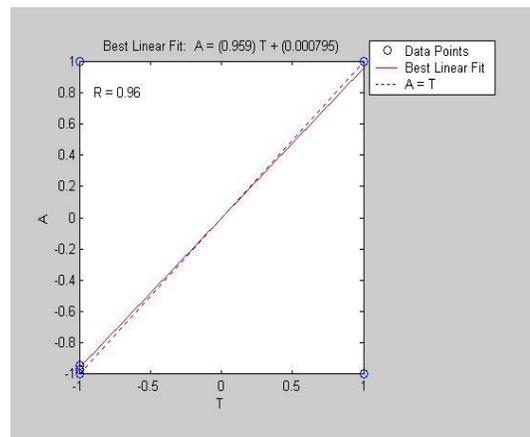
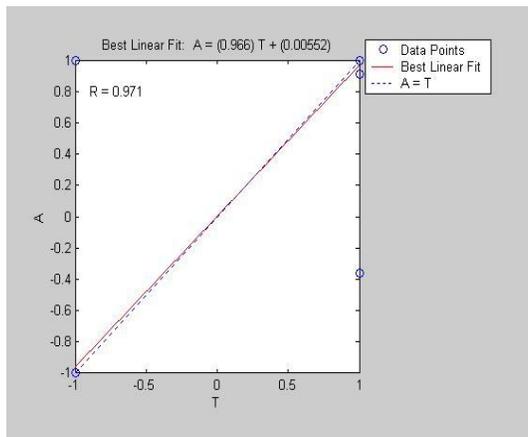


Figura 4.1: Regresiones para el lote de prueba 1.1

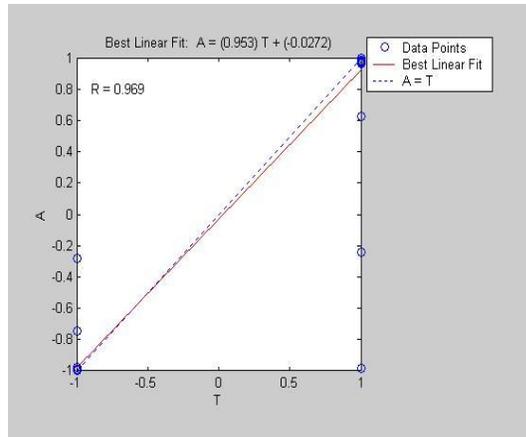
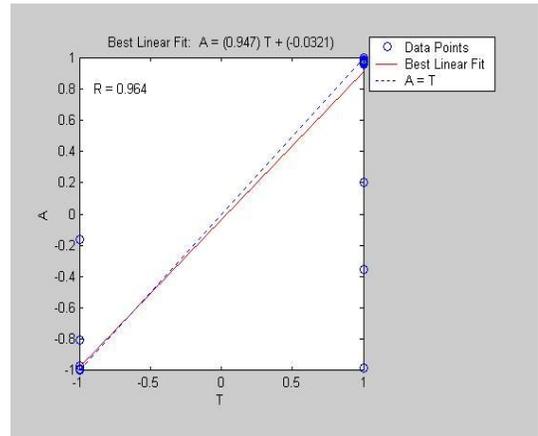
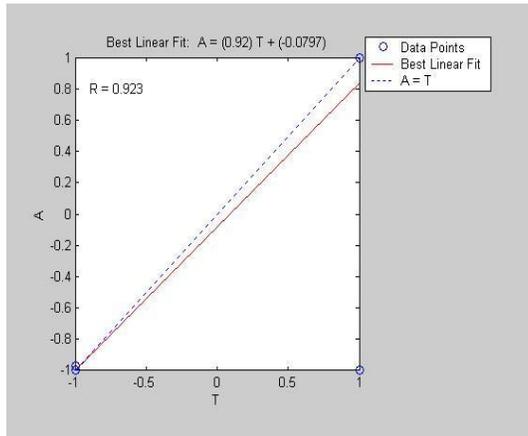
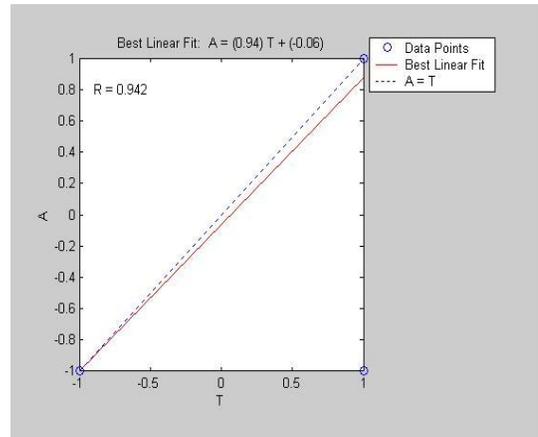
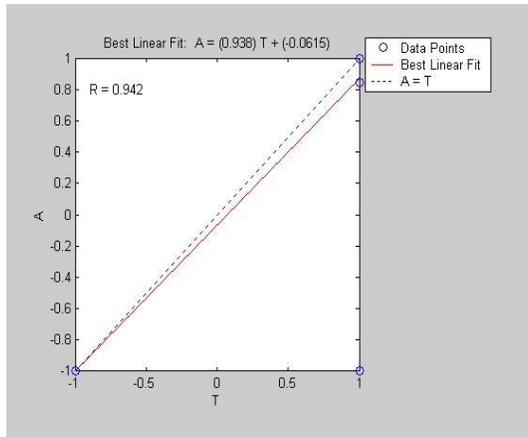


Figura 4.2: Regresiones para el lote de prueba 1.2

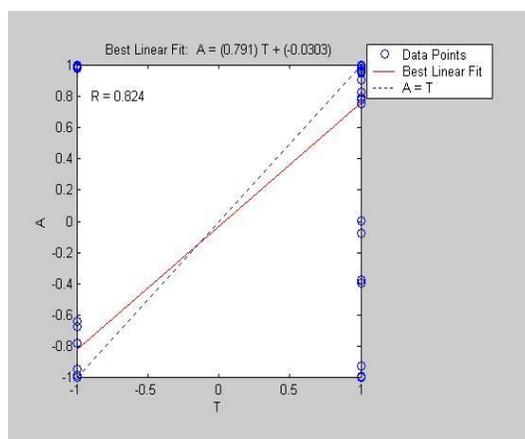
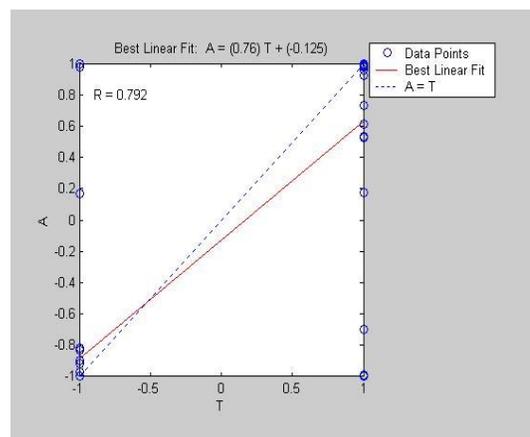
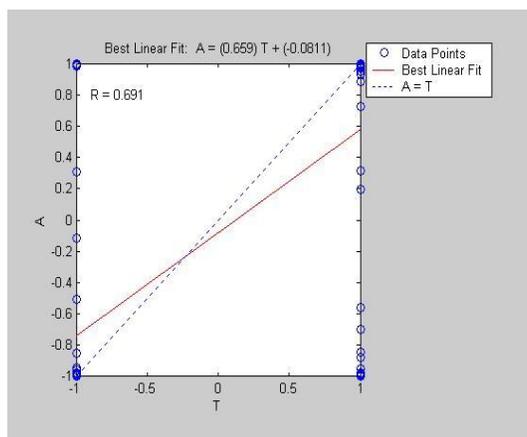
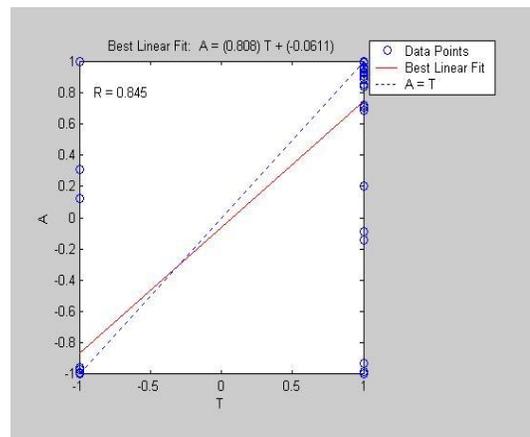
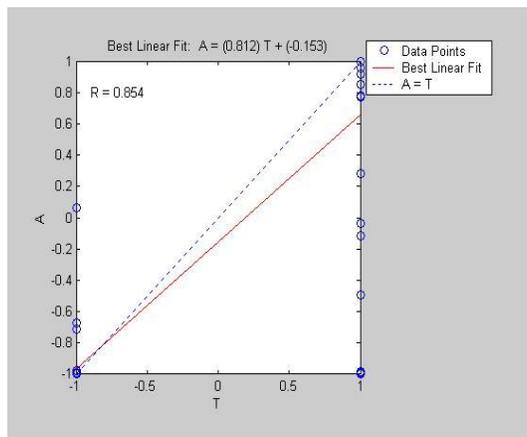


Figura 4.3: Regresiones para el lote de prueba 1.3

la configuración correspondiente a la inicialización 4. Para el lote del 901 al 1000 la mejor configuración está asociada con la inicialización número 4. El cuadro siguiente contiene los tres mejores resultados para los lotes de prueba:

| | Lote prueba 1-100 | Lote prueba 501-600 | Vectores prueba 901-1000 |
|----------|----------------------|------------------------|--------------------------|
| Aciertos | 100 % | 97 % | 96 % |
| m | 0.9982 | 0.9334 | 0.9204 |
| b | 1.8×10^{-3} | -1.13×10^{-2} | -2.2×10^{-2} |
| r | 0.9998 | 0.9405 | 0.9373 |
| timr | 14.5 | 15.78 | 12.17 |

Cuadro 4.4: Mejores Configuraciones para clasificación 1

El promedio de los porcentajes de acierto para esta clasificación es de 97.67 %. Además, se resalta que en esta clasificación hay un lote de prueba para el cual el número de aciertos es total.

4.2. Clasificación 2

En este problema se discrimina entre registros tipo LPS y VTA para la estación CR2R. La arquitectura de red 99-3-1 arrojó los mejores resultados.

Para presentar los resultados correspondientes a la clasificación 2 se hace las siguientes convenciones:

Lote de prueba 2.1. Desde el vector 1 hasta el 100 de la matriz que contiene el total de los vectores

Lote de prueba 2.2. Desde el vector 5011 hasta el 600 de la matriz que contiene el total de los vectores.

Lote de prueba 2.3. Desde el vector 901 hasta el 1000 de la matriz que contiene el total de los vectores

Los resultados para cada inicialización y lote de prueba son:

| | Init 1 | Init 2 | Init 3 | Init 4 | Init 5 |
|----------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Aciertos | 97 % | 93 % | 96 % | 96 % | 97 % |
| m | 0.9319 | 0.8685 | 0.9265 | 0.9169 | 0.9352 |
| b | 6.81×10^{-2} | 1.32×10^{-1} | 7.35×10^{-2} | 8.30×10^{-2} | 6.48×10^{-2} |
| r | 0.9387 | 0.8807 | 0.9332 | 0.9226 | 0.9407 |
| time | 18.08 | 13.03 | 14.47 | 11.77 | 12.19 |

Cuadro 4.5: Resultados lote de prueba 2.1

En la figura 4.4 se presentan las rectas obtenidas con la aplicación del método de regresión lineal ordenadas por inicializaciones (Init) de izquierda a derecha.

| | Init 1 | Init 2 | Init 3 | Init 4 | Init 5 |
|----------|------------------------|------------------------|------------------------|------------------------|------------------------|
| Aciertos | 97 % | 97 % | 96 % | 98 % | 98 % |
| m | 0.9385 | 0.0.94 | 0.9198 | 0.9466 | 0.9529 |
| b | -6.15×10^{-2} | -6.00×10^{-2} | -7.97×10^{-2} | -3.21×10^{-2} | -2.72×10^{-7} |
| r | 0.9416 | 0.9417 | 0.9229 | 0.9643 | 0.9691 |
| time | 18.91 | 15.11 | 12.67 | 39.04 | 115.11 |

Cuadro 4.6: Resultados lote de prueba 2.2

En la figura 4.5 se presentan las rectas obtenidas con la aplicación del método de regresión lineal ordenadas por inicializaciones (Init) de izquierda a derecha.

| | Init 1 | Init 2 | Init 3 | Init 4 | Init 5 |
|----------|-----------------------|------------------------|------------------------|------------------------|------------------------|
| Aciertos | 94 % | 91 % | 91 % | 92 % | 92 % |
| m | 0.9676 | 0.8252 | 0.8185 | 0.8325 | 0.8324 |
| b | 5.40×10^{-3} | -1.46×10^{-2} | -1.40×10^{-1} | -7.71×10^{-2} | -1.07×10^{-2} |
| r | 0.8785 | 0.8436 | 0.8284 | 0.859 | 0.8621 |
| time | 14.28 | 13.15 | 15.72 | 116.82 | 49.14 |

Cuadro 4.7: Resultados lote de prueba 2.3

En la figura 4.6 se presentan las rectas obtenidas con la aplicación del método de regresión lineal ordenadas por inicializaciones (Init) de izquierda a derecha.

Escogemos nuevamente para cada lote de prueba la configuración de pesos que mejores resultados dio. Para este caso las configuraciones que mejores resultados arrojan son las asociadas a las inicializaciones 5, 5 y 1 para cada lote de prueba respectivamente.

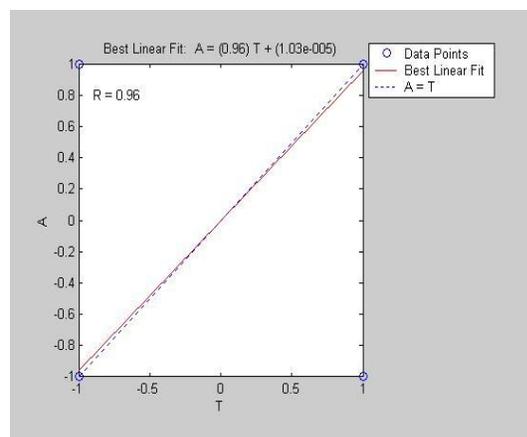
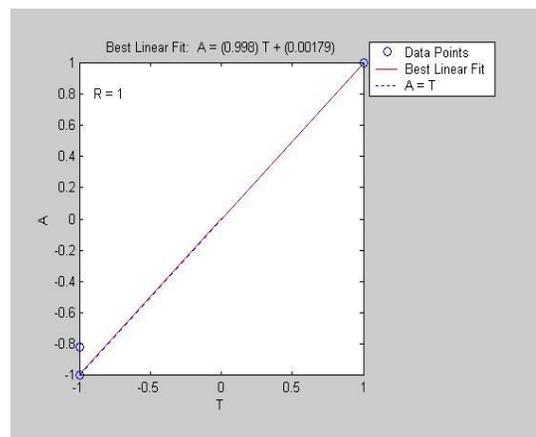
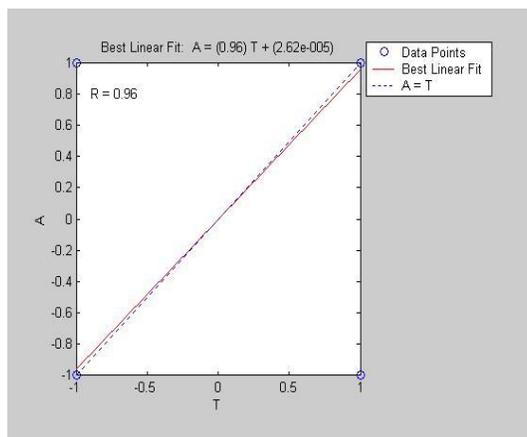
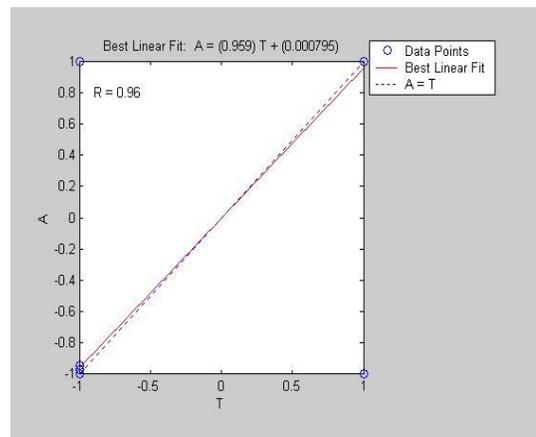
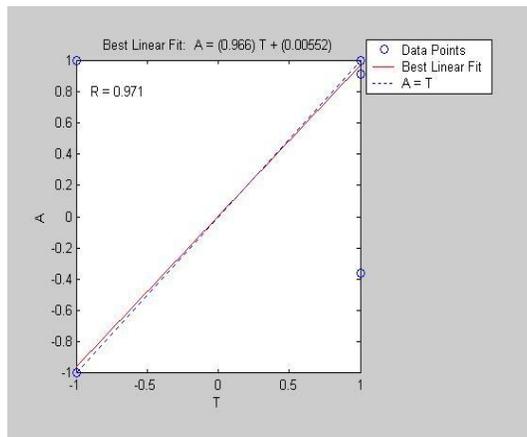


Figura 4.4: Regresiones para el lote de prueba 2.1

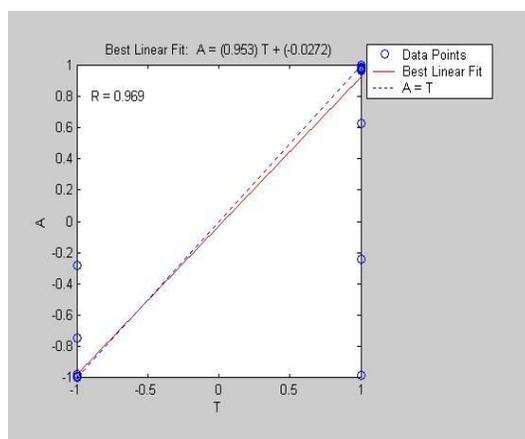
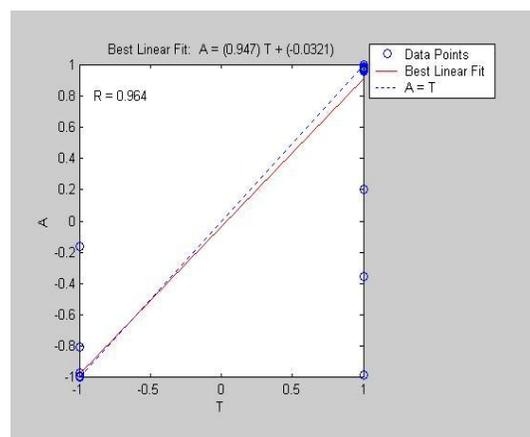
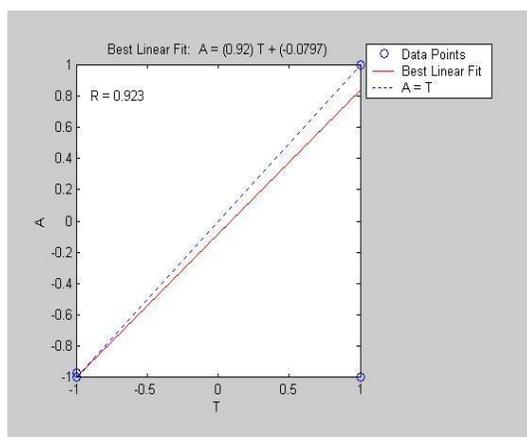
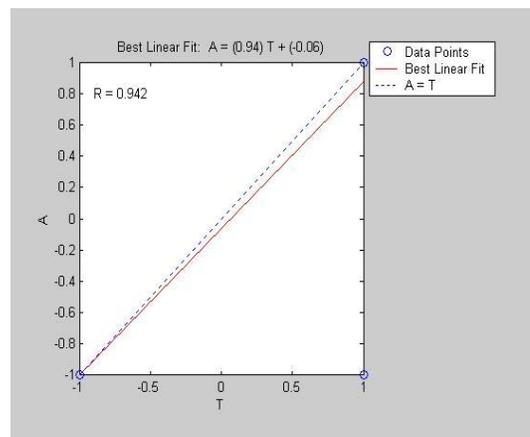
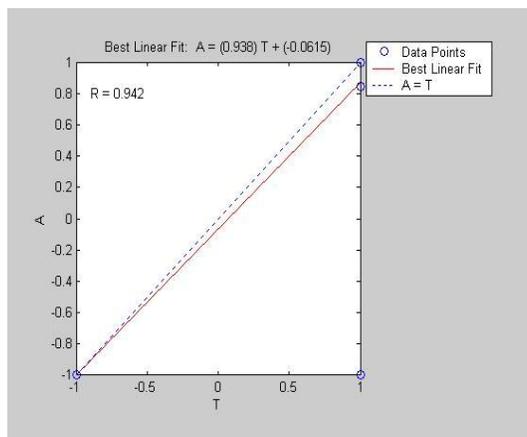


Figura 4.5: Regresiones para el lote de prueba 2.2

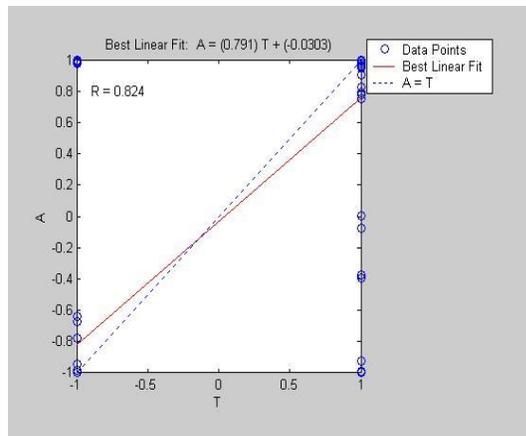
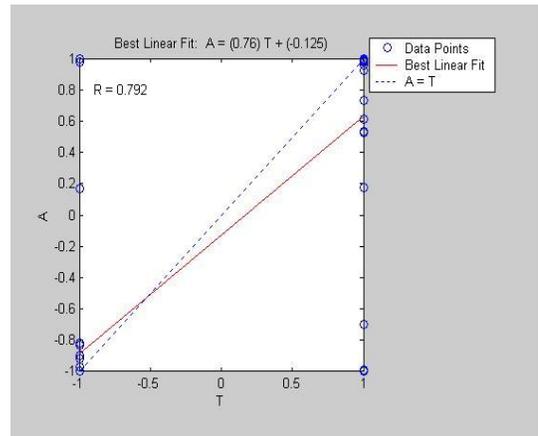
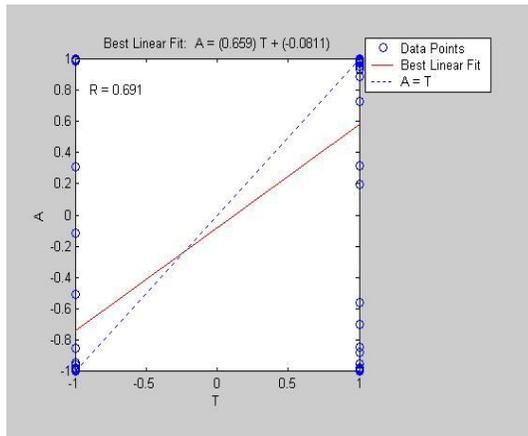
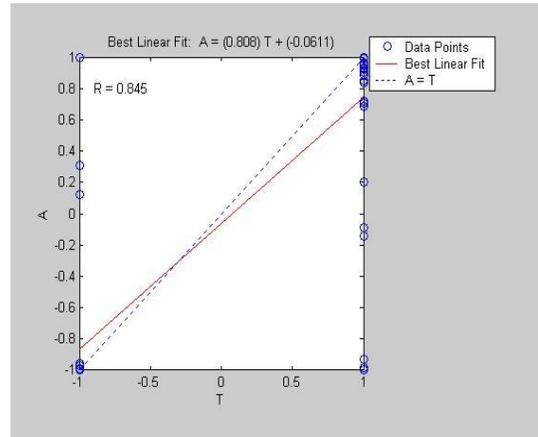
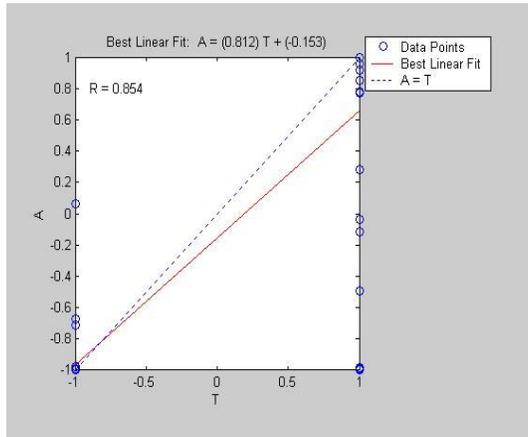


Figura 4.6: Regresiones para el lote de prueba 2.3

El cuadro siguiente resume los tres mejores resultados para los lotes de prueba:

| | Lote prueba 1-100 | Lote prueba 501-600 | Vectores prueba 901-1000 |
|----------|-----------------------|------------------------|--------------------------|
| Aciertos | 97 % | 98 % | 94 % |
| m | 0.9352 | 0.9529 | 0.9676 |
| b | 6.48×10^{-2} | -2.72×10^{-2} | 5.40×10^{-3} |
| r | 0.9407 | 0.9691 | 0.8785 |
| timr | 12.19 | 115.11 | 15.35 |

Cuadro 4.8: Mejores Configuraciones para clasificación 2

El promedio de los porcentajes para esta clasificación es de 96.33 % en aciertos.

4.3. Clasificación 3

En este problema se discrimina entre registros tipo LPS y HBD para la estación ANGV. La arquitectura de red 99-3-1 arrojó los mejores resultados.

Para presentar los resultados correspondientes a la clasificación 3 se hace las siguientes convenciones:

Lote de prueba 3.1. Desde el vector 1 hasta el 100 de la matriz que contiene el total de los vectores

Lote de prueba 3.2. Desde el vector 501 hasta el 600 de la matriz que contiene el total de los vectores.

Lote de prueba 3.3. Desde el vector 901 hasta el 1000 de la matriz que contiene el total de los vectores

Los resultados para cada inicialización y lote de prueba son los siguientes:

| | Init 1 | Init 2 | Init 3 | Init 4 | Init 5 |
|----------|-----------------------|-----------------------|-----------------------|-----------------------|------------------------|
| Aciertos | 92 % | 87 % | 89 % | 88 % | 85 % |
| m | 0.8285 | 0.7254 | 0.7383 | 0.7315 | 0.6997 |
| b | 5.11×10^{-2} | 2.73×10^{-2} | 0.05×10^{-2} | 1.26×10^{-2} | 12.34×10^{-2} |
| r | 0.8546 | 0.7455 | 0.7578 | 0.7596 | 0.7387 |
| time | 98.25 | 97.59 | 97.62 | 97.63 | 97.14 |

Cuadro 4.9: Resultados lote de prueba 3.1

En la figura 4.7 se presentan las rectas obtenidas con la aplicación del método de regresión lineal ordenadas por inicializaciones (Init) de izquierda a derecha.

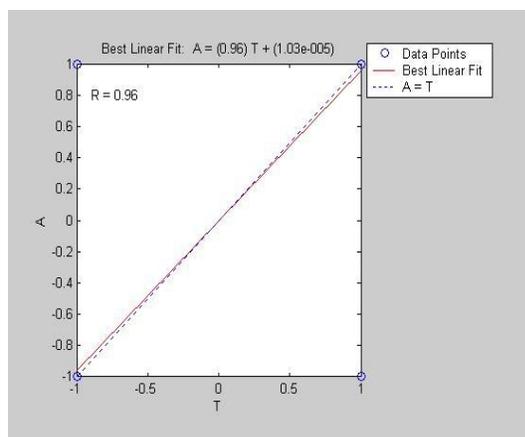
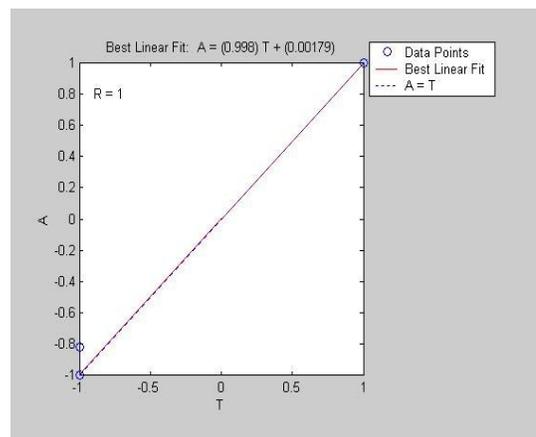
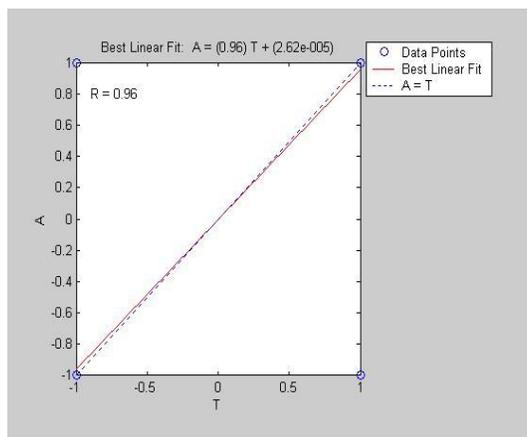
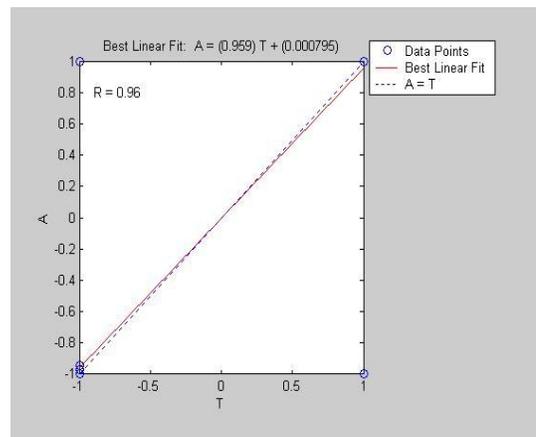
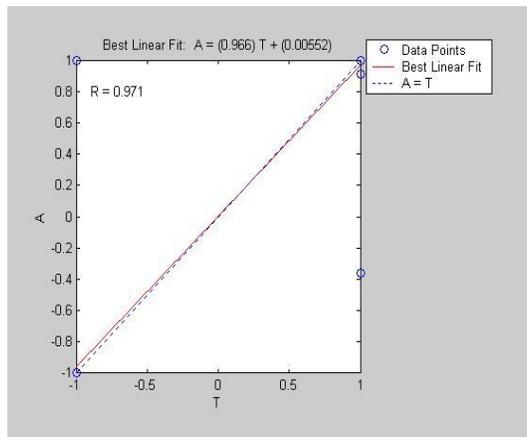


Figura 4.7: Regresiones para el lote de prueba 3.1

| | Init 1 | Init 2 | Init 3 | Init 4 | Init 5 |
|----------|------------------------|-------------------------|------------------------|-------------------------|-------------------------|
| Aciertos | 94 % | 94 % | 94 % | 95 % | 96 % |
| m | 0.8779 | 0.8849 | 0.8742 | 0.8814 | 0.8869 |
| b | -8.84×10^{-2} | -11.42×10^{-2} | -8.20×10^{-2} | -10.39×10^{-2} | -11.23×10^{-7} |
| r | 0.9006 | 0.9379 | 0.9231 | 0.9405 | 0.9206 |
| time | 59.2 | 59 | 118.5 | 58.9 | 59.25 |

Cuadro 4.10: Resultados lote de prueba 3.2

En la figura 4.8 se presentan las rectas obtenidas con la aplicación del método de regresión lineal ordenadas por inicializaciones (Init) de izquierda a derecha.

| | Init 1 | Init 2 | Init 3 | Init 4 | Init 5 |
|----------|-------------------------|------------------------|------------------------|-------------------------|------------------------|
| Aciertos | 90 % | 91 % | 84 % | 89 % | 90 % |
| m | 0.81 | 0.8083 | 0.6594 | 0.7604 | 0.7909 |
| b | -15.00×10^{-2} | -6.11×10^{-2} | -8.11×10^{-2} | -12.54×10^{-2} | -3.03×10^{-2} |
| r | 0.85 | 0.8453 | 0.6913 | 0.7924 | 0.8242 |
| time | 78.90 | 79.37 | 79.31 | 79.44 | 79.59 |

Cuadro 4.11: Resultados lote de prueba 3.3

En la figura 4.9 se presentan las rectas obtenidas con la aplicación del método de regresión lineal ordenadas por inicializaciones (Init) de izquierda a derecha.

Las configuraciones con mejores resultados para esta fase son la asociadas a las inicializaciones 1, 5 y 2 para cada lote de entrenamiento. Las características de los mejores resultados en las pruebas anteriores son las siguientes:

| | Lote prueba 1-100 | Lote prueba 501-600 | Vectores prueba 901-1000 |
|----------|-----------------------|-------------------------|--------------------------|
| Aciertos | 92 % | 96 % | 91 % |
| m | 0.8285 | 0.8869 | 0.8083 |
| b | 5.11×10^{-2} | -11.23×10^{-2} | -84.53×10^{-2} |
| r | 0.8546 | 0.9205 | 0.8453 |
| timr | 98.25 | 59.25 | 79.37 |

Cuadro 4.12: Mejores Configuraciones para clasificación 3

Tomando el promedio de los porcentajes para esta clasificación es de 93.00 % en aciertos.

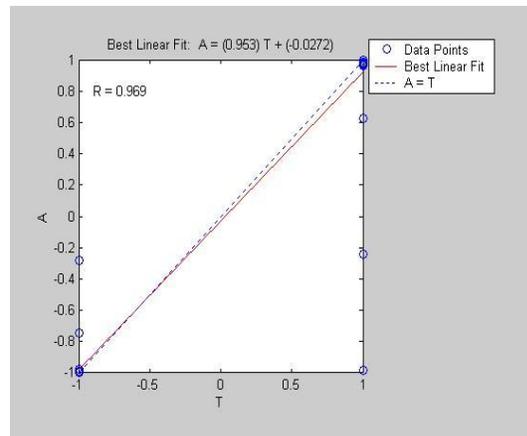
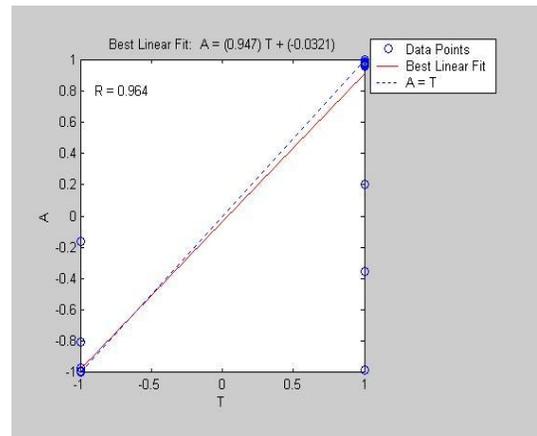
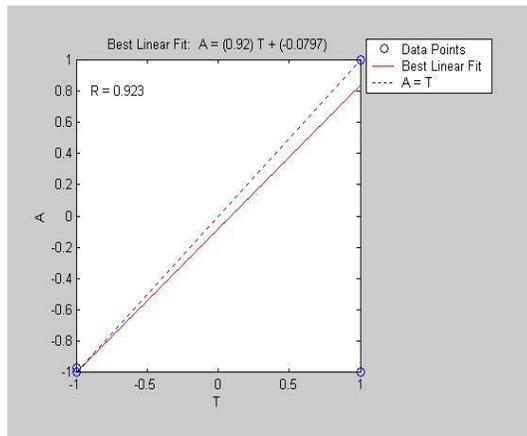
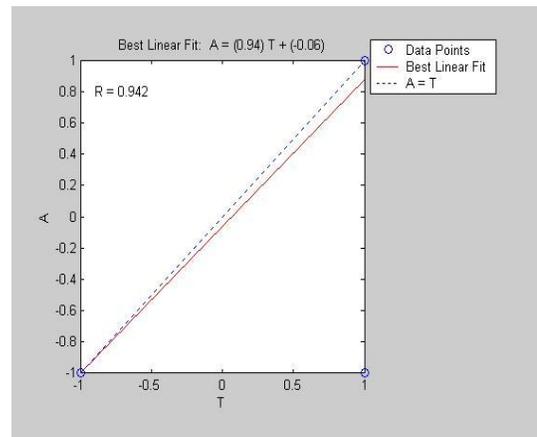
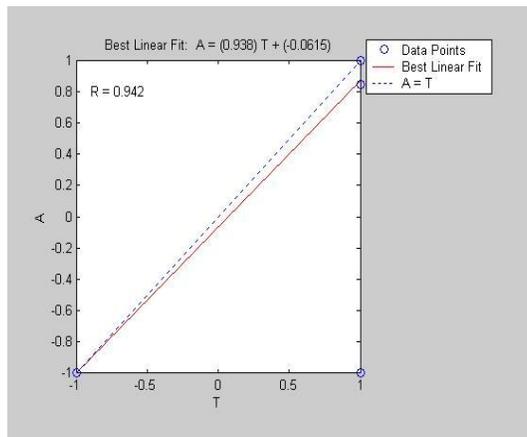


Figura 4.8: Regresiones para el lote de prueba 3.2

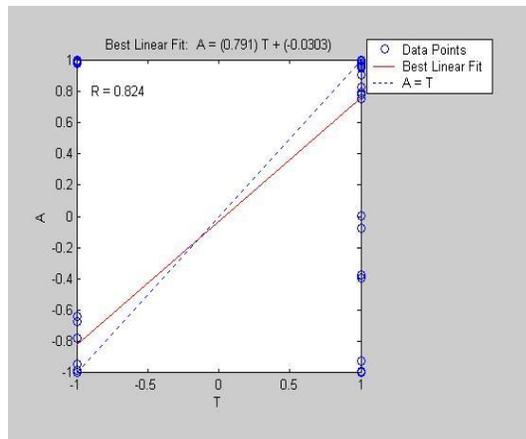
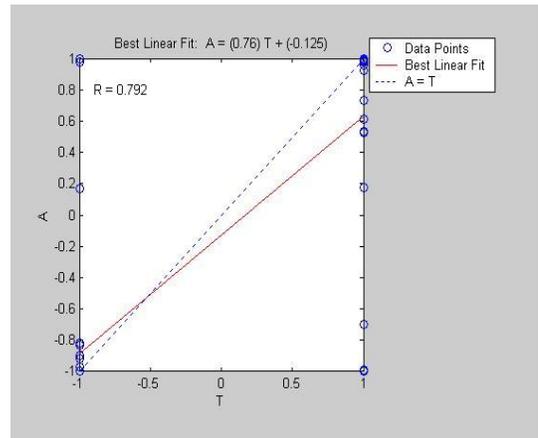
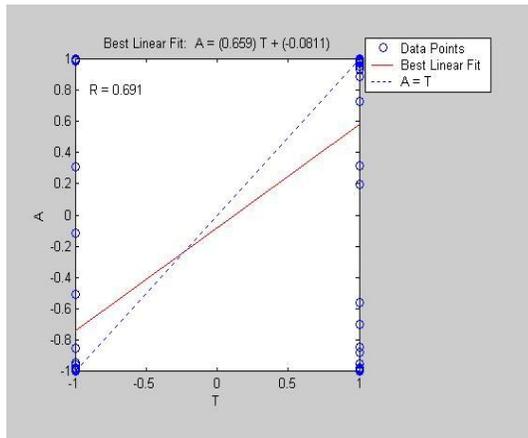
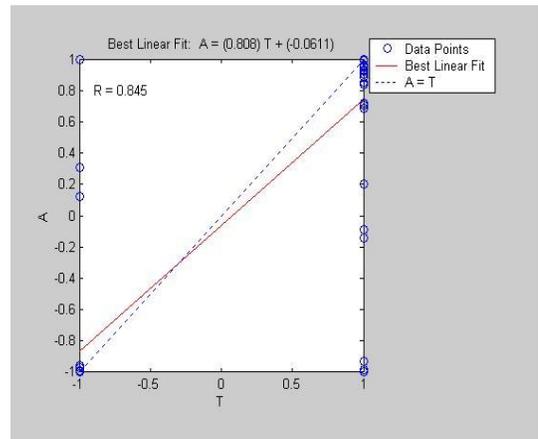
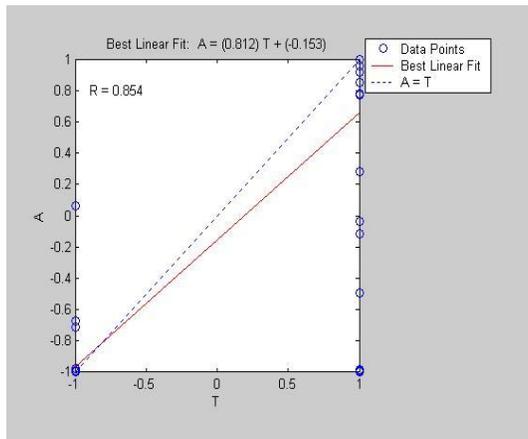


Figura 4.9: Regresiones para el lote de prueba 3.3

CONCLUSIONES

La clasificación entre diferentes tipos de registros sísmicos puede llevarse a cabo mediante el uso de Redes Neuronales Artificiales tipo Perceptrón Multicapa.

La representación de las señales mediante la técnica de los LPC entrega buenos resultados, extrayendo características robustas de cada señal. Esto se hace evidente cuando el Perceptrón utilizado reconoce fácilmente las características comunes a un mismo tipo de eventos, obteniendo los mejores resultados con un Perceptrón de arquitectura simple.

Para las 3 clasificaciones tratadas en este trabajo, la mejor configuración es 99-3-1. El número de neuronas en la capa oculta es bastante pequeño, lo cual asegura que durante el entrenamiento la red no experimenta una situación de sobreajuste, esto a su vez garantiza una buena generalización de la red.

El promedio de aciertos de la clasificación entre LPS y VTA para la estación ANGV es de 97.67 %, entre LPS y VTA para la estación CR2R es de 96.33 % y entre LPS y HBD para la estación ANGV es de 93 %.

Con los resultados anteriores se satisfacen las expectativas que inicialmente se tenían a cerca del uso de este tipo de herramientas.

RECOMENDACIONES

La metodología utilizada en este trabajo puede ser aplicada fácilmente a la clasificación entre más de dos tipos de eventos.

Ya que algunos tipos de eventos como el Tremor volcánico tienen duraciones mayores a la que en este trabajo se considera es necesario aumentar el tamaño de la ventana que contiene el evento y que es sometido a la extracción de los LPC.

Se sugiere realizar un tratamiento previo de los registros, utilizando filtros apropiados, con el fin de minimizar ruido.

Se recomienda la aplicación de diferentes algoritmos de entrenamiento para el proceso de aprendizaje de la red. Es posible que alguno de ellos brinde mejores resultados.

Bibliografía

- [1] DEL PEZZO, Edoardo. ESPOSITO, Anna. GIUDICE, Flora. MARINARO, Maria. MARTINI, Marcello. SCARPETTA, Silvia. Discrimination of earthquakes and underwater explosions using neuronal networks.
- [2] 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.
- [3] ISLAM, Tamaña. Interpolation of Linear Prediction Coeficients for speech coding; McGill University. Montreal Canada. 2000.
- [4] HAYKIN, Simon. Neuronal Networks; McMaster University. 1994.
- [5] DEL BRIO, Bonifacio. SANZ,Alfredo. Redes Neuronales y Sistemas Difusos; Universidad de Zaragoza. 2002.
- [6] PROAKIS, John. MANOLAKIS, Dimitris. Tratamiento digital de señales. 1998.
- [7] HAYKIN, Simon. Van Veen, Barry. Señales y Sistemas,2001
- [8] DEMUTH, Howard. BEALE, Mark. Neuronal Network Toolbox, MATLAB 6.0. 2002.

ANEXOS

Anexo A

Se entrega en medio magnético las tablas que contienen los resultados individuales para cada registro en la totalidad de pruebas en las 3 clasificaciones.

Anexo B

Se entrega en medio magnético las rutinas:

- estrac3
- autocolp
- autocovt
- autocohb

Igualmente en medio magnético están las 15 RNA con sus respectivas configuraciones.