

**CLASIFICACIÓN DE SEÑALES ELECTROMIOGRÁFICAS (EMG) PARA UNA
PRÓTESIS DE MANO**



**CARLOS MARIO BELALCÁZAR SANDOVAL
ZULLY VIVIANA RENGIFO VARILA**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL
LÍNEA DE CONTROL DE PROCESOS
POPAYÁN
2007**

**CLASIFICACIÓN DE SEÑALES ELECTROMIOGRÁFICAS (EMG) PARA UNA
PRÓTESIS DE MANO**



**CARLOS MARIO BELALCAZAR SANDOVAL
ZULLY VIVIANA RENGIFO VARILA**

**Trabajo de grado presentado como requisito para obtener el título de
Ingeniero en Automática Industrial**

**Director
Ing. JAIME OSCAR DÍAZ CHAVEZ**

**UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE ELECTRÓNICA, INSTRUMENTACIÓN Y CONTROL
LÍNEA DE CONTROL DE PROCESOS
POPAYÁN
2007**

AGRADECIMIENTOS

Agradezco a Dios por darme una familia ejemplar; mi madre, mi padre y mi hermano quienes me apoyaron incondicionalmente y motivaron a seguir adelante con mis estudios. También le doy gracias a las personas que intervinieron en este proceso de educación, a los docentes, a mis compañeros de estudio, mi compañera de tesis y mi novia quienes estuvieron a mi lado en los buenos y malos momentos.

Carlos Mario Belalcázar Sandoval

A Dios, ante todo, por acompañarme y guiarme no solo en el desarrollo de este proyecto si no durante toda mi vida. A mis padres, mis hermanos y mi abuela por creer en mí y apoyarme en todo momento. A mi compañero de tesis y amigos por animarme a seguir siempre adelante. A los profesores Harold Romo y Jaime Díaz por su colaboración.

Zully Viviana Rengifo Varila

CONTENIDO

	Pág.
RESUMEN.....	1
INTRODUCCIÓN.....	1
1. RECONOCIMIENTO DE PATRONES EN SEÑALES.....	3
1.1 EXTRACCIÓN DE CARACTERÍSTICAS.....	4
1.1.1 Análisis temporal.....	5
1.1.2 Análisis tiempo – frecuencia	7
1.2 REDUCCIÓN DE DIMENSIONALIDAD.....	10
1.2.1 Reducción por selección de características.....	11
1.2.2 Reducción por proyección de características	11
1.3 CLASIFICACIÓN DE PATRONES.....	12
1.3.1 Formas de clasificación	14
1.3.2 Consideraciones para la evaluación de un clasificador	16
2. FUNDAMENTOS TEÓRICOS DE LOS CLASIFICADORES	18
2.1 PERCEPTRÓN MULTICAPA (MLP).....	19
2.2 REDES NEURONALES PROBABILÍSTICAS (PNN)	26
2.3 MAQUINA DE VECTORES DE SOPORTE (SVM).....	30
2.3.1 Clasificación utilizando SVM.....	31
2.3.2 Análisis del caso linealmente separable	32
2.3.3 Solución al caso linealmente no separable con variables de relajación	37
2.3.4 Máquinas de vectores soporte no lineal.....	39
2.3.5 SVM multiclase	41
2.4 REDES FUZZY ARTMAP	44
2.4.1 Fuzzy ART	44
2.4.2 Red Fuzzy ARTMAP.....	48
3. CLASIFICADORES IMPLEMENTADOS.	52
3.1 DESCRIPCIÓN DE LA BASE DE DATOS.....	53

3.1.1	Descripción de los vectores de características	55
3.1.2	Clases de salida.	56
3.1.3	Vectores de entrenamiento.....	56
3.1.4	Construcción de los clasificadores.....	57
3.1.5	Vectores de validación.....	58
3.1.6	Diagrama en bloques.....	58
3.2	CLASIFICADOR CON LA RED PERCEPTRÓN MULTICAPA (MLP).....	59
3.3	CLASIFICADOR CON LA RED NEURONAL PROBABILÍSTICA (PNN) ..	61
3.4	CLASIFICADOR CON MÁQUINAS DE VECTORES SOPORTE (SVM) ..	62
3.4.1	Asumiendo datos linealmente separables	64
3.4.2	Asumiendo datos linealmente no separables	65
3.5	CLASIFICADOR CON UNA RED FUZZY ARTMAP.....	67
4	PRUEBAS Y EVALUACIÓN DE LOS CLASIFICADORES.....	68
4.1	PRUEBAS.....	68
4.1.1	Perceptrón multicapa (MLP)	69
4.1.2	Red neuronal probabilística (PNN)	71
4.1.3	SVM asumiendo separación de datos lineal.	73
4.1.4	SVM con función kernel polinomial	74
4.1.5	SVM con kernel función de base radial (RBF)	76
4.1.6	Fuzzy ARTMAP	79
4.2	EVALUACION.....	80
4.2.1	Exactitud o precisión.....	81
4.2.2	Tiempo de clasificación.....	83
4.2.3	Potencia.....	86
4.2.4	Flexibilidad.....	88
5	CONCLUSIONES Y RECOMENDACIONES.....	90
5.1	CONCLUSIONES	90
5.2	RECOMENDACIONES PARA TRABAJOS FUTUROS	93
	BIBLIOGRAFIA.....	94

LISTA DE FIGURAS

	Pág.
Figura 1. Reconocimiento de patrones de las señales EMG.	3
Figura 2. a) Problema separable linealmente, b) Problema no separable linealmente.....	15
Figura 3. Matriz de confusión.....	16
Figura 4. Arquitectura típica de una MLP.....	19
Figura 5. (a) Función logarítmica sigmoidea, (b) Función tangente sigmoidea.....	21
Figura 6. Esquema básico de una PNN.....	27
Figura 7. Función de transferencia de una neurona de base radial.....	29
Figura 8. Hiperplanos que separan correctamente un conjunto de datos.....	32
Figura 9. Hiperplano lineal clasificador para el caso no separable.	38
Figura 10. SVM no lineal representada en una nueva dimensión alta.	40
Figura 11. Esquema de la red Fuzzy ART.....	45
Figura 12. Esquema general de una red Fuzzy ARTMAP.....	49
Figura 13. Interfaz grafica del programa CLASIFICACION DE SEÑALES EMG. ...	52
Figura 14. Matriz de vectores de características en 3D.....	54
Figura 15. Conformación del vector de características para una muestra.	55
Figura 16. Tipos de movimientos a clasificar.....	56
Figura 17. Matriz de vectores de características en 2D para la entrada del clasificador.....	57
Figura 18. Esquema del proceso de clasificación.....	58

LISTA DE GRAFICAS

	Pág.
Grafica 1. Resultados de la clasificación. Técnica: DT – DWT, Método: MLP	70
Grafica 2. Resultados de la clasificación. Técnica: DT – WPT, Método: MLP	71
Grafica 3. Resultados de la clasificación. Técnica: DT – DWT, Método: PNN.....	72
Grafica 4. Resultados de la clasificación. Técnica: DT – WPT, Método: PNN.....	73
Grafica 5. Resultados de la clasificación. Técnica: DT - DWT y DT - WPT, Método: SVM LINEAL.....	74
Grafica 6. Resultados de la clasificación. Técnica: DT – DWT, Método: SVM POLINOMIAL.....	75
Grafica 7. Resultados de la clasificación. Técnica: DT – WPT, Método: SVM POLINOMIAL.....	76
Grafica 8. Resultados de la clasificación. Técnica:DT-DWT, Método:SVM RBF ..	77
Grafica 9. Resultados de la clasificación. Técnica:DT-WPT, Método:SVM RBF ..	78
Grafica 10. Resultados de la clasificación. Técnica: DT-DWT, Método: Fuzzy ARTMAP	79
Grafica 11. Resultados de la clasificación. Técnica: DT – WPT, Método: Fuzzy ARTMAP	80
Grafica 12. Comparación de la precisión de los clasificadores con la técnica DT_DWT.....	81
Grafica 13. Comparación de la precisión de los clasificadores con la técnica DT_WPT	82
Grafica 14. Comparación de la velocidad promedio de los clasificadores con la técnica DT_DWT.....	84
Grafica 15. Comparación de la velocidad promedio de los clasificadores con la técnica DT_WPT.....	85

Grafica 16. Comparación de la potencia promedio de los clasificadores con la técnica DT_DWT.....86

Grafica 17. Comparación de la potencia promedio de los clasificadores con la técnica DT_WPT.....87

Grafica 18. Comparación de la flexibilidad de los clasificadores con la técnica DT_DWT.....88

Grafica 19. Comparación de la flexibilidad de los clasificadores con la técnica DT_WPT.....89

LISTA DE TABLAS

	Pág.
Tabla 1. Porcentajes de error de clasificación. Técnica: DT-DWT, Método: MLP.	70
Tabla 2. Porcentajes de error de clasificación. Técnica: DT- WPT, Método: MLP.	71
Tabla 3. Porcentajes de error de clasificación. Técnica: DT-DWT, Método: PNN.	72
Tabla 4. Porcentajes de error de clasificación. Técnica: DT-WPT, Método: PNN.	73
Tabla 5. Porcentajes de error de clasificación. Técnica: DT – WPT y DT – WPT, Método: SVM LINEAL.....	74
Tabla 6. Porcentajes de error de clasificación. Técnica: DT – DWT, Método: SVM POLINOMIAL.....	75
Tabla 7. Porcentajes de error de clasificación. Técnica: DT – WPT, Método: SVM POLINOMIAL.....	76
Tabla 8. Porcentajes de error de clasificación. Técnica: DT – DWT, Método: SVM RBF.....	77
Tabla 9. Porcentajes de error de clasificación. Técnica: DT – WPT, Método: SVM RBF.....	78
Tabla 10. Porcentajes de error de clasificación. Técnica: DT – DWT, Método: Fuzzy ARTMAP	79
Tabla 11. Porcentajes de error de clasificación. Técnica: DT – WPT, Método: Fuzzy ARTMAP	80
Tabla 12. Comparación de la precisión de los clasificadores con la técnica DT_DWT.....	82
Tabla 13. Comparación de la precisión de los clasificadores con la técnica DT_WPT.....	82
Tabla 14. Comparación de la velocidad promedio de los clasificadores con la técnica DT_DWT.....	84

Tabla 15. Comparación de la velocidad promedio de los clasificadores con la técnica DT_WPT.....85

Tabla 16. Comparación de la potencia promedio de los clasificadores con la técnica DT_DWT.....86

Tabla 17. Comparación de la potencia promedio de los clasificadores con la técnica DT_WPT.....87

Tabla 18. Comparación de la flexibilidad de los clasificadores con la técnica DT_DWT.....89

Tabla 19. Comparación de la flexibilidad de los clasificadores con la técnica DT_WPT.....89

RESUMEN

En este documento se presentan cuatro técnicas inteligentes, para la clasificación de las señales Electromiográficas (EMG) que se originan con ejecución de cuatro movimientos básicos de mano, (mano en reposo, agarre fino, agarre grueso y mano abierta), realizados por seis pacientes diferentes. Estos métodos son: red neuronal probabilística (PNN), perceptrón multicapa (MLP), máquinas de vectores soporte (SVM) y fuzzy ARTMAP,

Se exponen los fundamentos teóricos de cada método de clasificación, precedido por una breve explicación sobre el proceso de reconocimiento de patrones y sus fases: extracción de características, reducción de dimensionalidad y clasificación en general. En sección de extracción de características se hace referencia a las características que forman parte de los vectores que se utilizan como datos de entrada del clasificador,

Posteriormente se presenta el proceso de creación de los clasificadores en donde se hace referencia a la base de datos de las señales EMG utilizada para la creación y la validación de los clasificadores y los parámetros finales utilizados en el clasificador; luego son comparados y evaluados de acuerdo a sus resultados y a cuatro propiedades (precisión, velocidad de clasificación, potencia y flexibilidad) definidas para tal fin.

PALABRAS CLAVES: señales electromiográficas, extracción de características, clasificación, red neuronal probabilística (PNN), perceptrón multicapa (MLP), máquinas de vectores soporte (SVM), fuzzy ARTMAP.

INTRODUCCIÓN

El número de personas discapacitadas en Colombia por la pérdida parcial o total de una o ambas extremidades superiores es considerable. Entre las causas principales de dicha discapacidad están: los accidentes laborales, de tránsito, domésticos y actos de violencia. Generalmente el tratamiento clínico que reciben estos pacientes se limita a cirugías de amputación y una pequeña asistencia psicológica postoperatoria. Sin embargo, las personas que han sufrido la amputación de una o ambas manos, además de los efectos sociológicos, experimentan consecuencias fisiológicas y psicológicas. Estos efectos descompensan el equilibrio del organismo deteriorando los grupos musculares asociados a la extremidad, disminuyen la autoestima y deterioran la relación con el entorno.

Un problema esencial, en estos casos, es cómo compensar, en lo posible y de la mejor manera, la falta de la extremidad superior en personas que han sido amputadas a la altura del antebrazo y de esta forma, mejorar la calidad de vida de los discapacitados. Un inicio para conseguir la solución a dicho problema es la adquisición y reconocimiento de patrones de las señales electromiográficas (EMG) provenientes de los músculos que están asociadas cualitativamente con el tipo de movimiento producido.

Gracias a que las señales electromiográficas (EMG) pueden ser generadas de manera voluntaria por las personas, representan una excelente opción para ser utilizadas en el control de prótesis inteligentes. El reconocimiento de patrones de dichas señales, requiere abordar tres fases: extracción de características, reducción de dimensionalidad, y clasificación de patrones de movimiento

La fase de clasificación, dentro del proceso de análisis de dichas señales ó reconocimiento de patrones, es un componente muy importante y puede considerarse como la tarea final de procesado, pues una vez definido el experimento, recogidas las señales y pre-procesadas convenientemente, el último paso es la implementación de un clasificador que pueda decidir a qué clase corresponde cada una de ellas.

Por esta razón y para contribuir en la búsqueda de que la persona discapacitada recupere algunas de sus capacidades físicas que le permitan desarrollar actividades cotidianas que le faciliten adaptarse a la sociedad, el principal objetivo de este documento ha sido realizar la clasificación de las señales EMG para una prótesis de mano para identificar la pertenencia o no de una señal EMG a uno de los cuatro tipos de movimiento básicos de mano (mano en reposo, agarre fino, agarre grueso y mano abierta), utilizando cuatro técnicas inteligentes de clasificación: una red neuronal probabilística (PNN), un perceptrón multicapa (MLP), máquinas de vectores soporte (SVM) y fuzzy ARTMAP.

A fin de desarrollar lo propuesto, en el capítulo 1 se realizará una descripción de las tres fases que componen el proceso de reconocimiento de patrones. En el capítulo 2, se exponen los fundamentos teóricos de los métodos de clasificación a implementar. En el capítulo 3, se presenta el proceso de creación de los clasificadores y los parámetros finales utilizados, además del origen y la descripción de los vectores de características híbridos utilizados para la creación y la validación de los clasificadores. En el capítulo 4, se muestran las pruebas efectuadas, los resultados obtenidos y se hace la evaluación de la implementación de los métodos. Finalmente se exponen las conclusiones y recomendaciones basadas en la experiencia del proyecto, para que puedan ayudar al desarrollo de futuros trabajos e investigaciones en esta área.

1. RECONOCIMIENTO DE PATRONES EN SEÑALES

En general los dispositivos de asistencia, como las prótesis, están compuestos por: la señal fisiológica de control (señal EMG), una etapa de procesamiento y control, en donde las señales fisiológicas son analizadas y convertidas en comandos de control, y el dispositivo de salida (prótesis, ortesis, estimulación FES, etc). Todas estas etapas están vinculadas por lazos de retroalimentación (sensores en los dispositivos de salida y algoritmos adaptativos en las etapas de procesamiento y control) [1].

Ya obtenidas las señales fisiológicas de control, en la etapa de procesamiento se hace un proceso de reconocimiento de patrones de señales que consta de tres grandes fases a implementar: la primera consiste en realizar un procesamiento de las señales, para extraer las características diferenciadoras atenuando el ruido presente, la segunda etapa corresponde a la reducción de dimensionalidad, fase que tiene como objetivo eliminar ciertos datos que no representan información significativa o eliminar aquellos datos que estén fuertemente correlacionados, y la tercera realiza la clasificación de las señales ó patrones a partir de las características obtenidas para determinar la clase a la que pertenece dicha señal. La relación existente entre estas etapas es mostrada en la figura 1 [1].

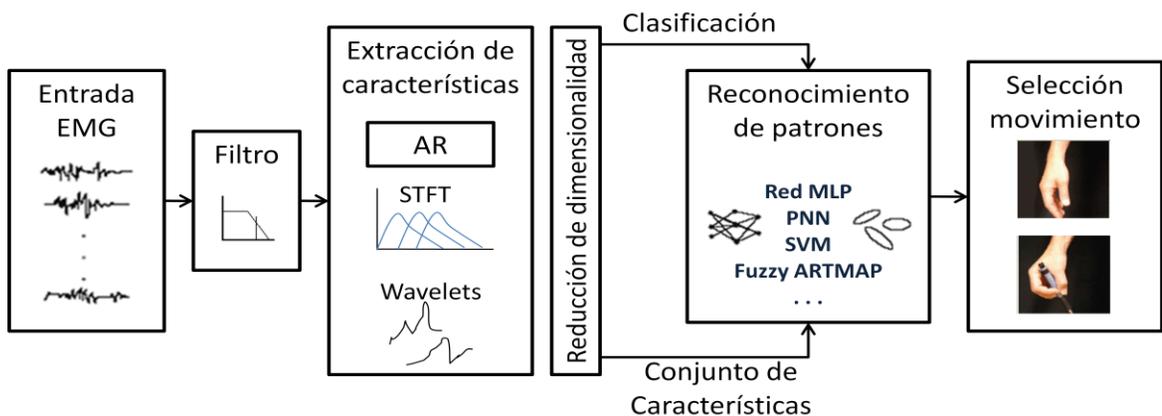


Figura 1. Reconocimiento de patrones de las señales EMG.

1.1 EXTRACCIÓN DE CARACTERÍSTICAS

La extracción de características es un proceso por el cual los atributos de las señales son calculados y recolectados en un formato de vector compacto. El objetivo de este paso es crear una representación manejable y significativa de la señal EMG original, con mira a maximizar el éxito potencial de la fase de clasificación y a su vez el rendimiento global del sistema. Un segundo objetivo de la fase de extracción de características es extraer la información útil comprimiendo los datos sin pérdida de información relevante, con objeto de reducir el número de variables de entrada para incrementar el desempeño de la fase de clasificación [2]. El éxito de la fase de extracción de características depende básicamente de la calidad de las señales medidas.

En el control mioeléctrico, los vectores de características que representan la información extraída de las señales EMG, son elegidos minimizando el error de control. Para lograr esto, el conjunto de características debe ser elegido de tal manera que se maximice la separación entre clases y minimice la distancia entre patrones de una misma clase. Así, la extracción de características es el punto principal del sistema de clasificación y es esencial para la identificación de comandos de movimiento. Sin embargo, el hecho de que las señales EMG sean no estacionarias, hace difícil la extracción de parámetros característicos que representen los eventos que se desean monitorear. Hasta el momento no se ha podido obtener un único parámetro que refleje a la perfección las características de la señal EMG medida. Debido a esto es necesario elegir un conjunto de características para ingresarlas a una etapa de clasificación de patrones [3].

En general los métodos para extracción de características son de gran importancia para la clasificación exitosa. Un sistema de clasificación exitosa depende no solo de la confianza, estabilidad y adaptabilidad del clasificador, sino también de la

efectividad, las estrategias o métodos de extracción y selección de características de los datos naturales.

Se pueden considerar dos formas para realizar la extracción de características: aproximación paramétrica y no paramétrica. En la aproximación paramétrica se parte de una función de distribución conocida, y se reduce el problema a estimar los parámetros que mejor ajusten las observaciones de la muestra. En la aproximación no paramétrica los parámetros calculados son independientes de la función de distribución. Dentro de la aproximación no – paramétrica se incluyen el análisis temporal, análisis espectral y análisis tiempo–frecuencia [1].

La aproximación paramétrica y el análisis espectral no serán abordados en este documento, debido a que los vectores de características utilizados en la implementación fueron creados a partir de análisis temporal y análisis tiempo – frecuencia. Estos dos temas serán abordados de una manera general, pues su estudio detallado no hace parte de los objetivos del proyecto. Sin embargo una explicación más detallada, en particular, de la transformada wavelet discreta (DWT) y wavelet packet (WPT) aplicada a la extracción de características EMG se encuentra en el anexo B y en [5].

1.1.1 Análisis temporal

Cuando se habla de una secuencia de valores observados a lo largo del tiempo, y por tanto ordenados cronológicamente, se le denomina serie temporal. En el análisis temporal la extracción de características identifica los atributos de las señales EMG naturales que caracterizan su estructura temporal relativa a una función muscular específica [6].

Las características en el dominio del tiempo son generalmente calculadas rápidamente, porque ellas no necesitan ninguna transformación. Algunas de esas

características han sido ampliamente investigadas y utilizadas en los trabajos referenciados en [1], [6], y [7].

Entre las características temporales tenemos:

1. *Valor Medio Absoluto (VMA)*: Este es un estimador del valor medio absoluto de la señal x_i , en el segmento i , el cual tiene una longitud de N muestras. Este puede ser expresado de la siguiente forma:

$$\bar{X}_i = \frac{1}{N} \sum_{k=1}^N |x_k| \quad (1.1)$$

Donde x_k es la muestra k -ésima en el segmento i .

2. *Valor Medio Absoluto de la Diferencia (VMAD)*: Este es el valor medio absoluto de la diferencia entre muestras adyacentes, k y $k+1$ y está definido por:

$$\Delta \bar{X}_i = \frac{1}{N-1} \sum_{k=1}^{N-1} |x_{k+1} - x_k| \quad (1.2)$$

3. *Cruces por cero (CC)*: Es una medida simple de frecuencia y puede ser obtenida por el número de veces que la onda o la señal cruza por cero. A fin de reducir los cruces por cero a consecuencia del ruido, se incluye un umbral ε . Al considerar dos muestras consecutivas x_k y x_{k+1} , CC es incrementado si:

$$\{x_k > 0 \text{ y } x_{k+1} < 0\} \text{ o } \{x_k < 0 \text{ y } x_{k+1} > 0\} \text{ y } |x_k - x_{k+1}| \geq \varepsilon \quad (1.3)$$

4. *Longitud de la Forma de Onda*: Es una característica que proporciona información sobre la complejidad de la forma de onda en cada segmento, ésta se calcula sobre un segmento de tiempo y se define como:

$$l_0 = \sum_{k=1}^N |\Delta x_k| \quad (1.4)$$

Donde $\Delta x_k = x_k - x_{k-1}$. Los valores resultantes indican una medida de la amplitud, frecuencia y duración de la forma de onda, todo esto en un solo parámetro.

5. *Cambio de pendiente*: Otra característica que puede dar idea del contenido de frecuencia de una señal puede ser el número de veces que cambia la pendiente de una señal. De nuevo, es recomendable seleccionar un umbral para reducir los cambios de pendiente inducidos por el ruido. Dadas tres muestras consecutivas x_{k-1} , x_k y x_{k+1} , el cambio de pendiente es incrementado si:

$$\begin{aligned} \{x_k > x_{k-1}\} \text{ y } \{x_k > x_{k+1}\} \quad \text{ó} \quad \{x_k < x_{k-1}\} \text{ y } \{x_k > x_{k+1}\} \text{ y} \\ |x_k - x_{k-1}| \geq \varepsilon \quad \text{ó} \quad |x_k - x_{k+1}| \geq \varepsilon \end{aligned} \quad (1.5)$$

1.1.2 Análisis tiempo – frecuencia

El análisis tiempo-frecuencia permite la localización conjunta de eventos en tiempo y frecuencia, siendo adecuado para el análisis de las señales no estacionarias como las EMG, las cuales se caracterizan por alta variabilidad del espectro en tiempos relativamente cortos [6], [7].

La Transformada wavelet (TW) surge como una alternativa para superar los problemas de resolución de la *Transformada de Fourier de Tiempo Corto* (*Short-Time Fourier Transform – STFT*), que es una herramienta de análisis frecuencial muy utilizada en el tratamiento de señales, pero que posee la gran desventaja de no facilitar información temporal. La TW, a diferencia de la STFT hace posible una buena representación de una señal tanto en tiempo como en frecuencia de forma simultánea, con lo que se puede determinar el intervalo de tiempo en el cual aparecen determinadas componentes espectrales [7], [8].

La TW pertenece a una serie de técnicas de análisis de señal denominadas comúnmente análisis multiresolución (AMR) que estudia la señal de tal forma que cada componente de frecuencia es analizado con una resolución diferente y descompone una serie de tiempo en diferentes *escalas temporales*. De esta manera se tiene que las *wavelets* funcionan como filtros (no lineales), los cuales permiten extraer componentes que contienen información en el dominio del tiempo y en el dominio de la frecuencia, éste último expresado en términos de *escalas* [9]. Su principal característica es que permite conocer qué frecuencias componen una señal en cada instante con las siguientes resoluciones [10]:

- Para las altas frecuencias consigue una buena resolución en el tiempo que permite su exacta localización temporal, aún a cambio de perder resolución frecuencial.
- Para las componentes de bajas frecuencias lo más relevante es conocer su frecuencia aún a costa de perder resolución temporal.

Existen diferentes tipos de *transformadas wavelet* (WT), entre las cuales se encuentran la *transformada wavelet continua* (CWT), la *transformada wavelet discreta* (DWT) y la *transformada wavelet packet* (WPT). La diferencia entre ellas radica principalmente en la forma en que los parámetros de translación y escala son discretizados [10]. Para más información sobre la DWT y la WPT remitirse al anexo B.

➤ **Características DWT y WPT**

Sobre la información extraída utilizando la transformada wavelet se calculan algunas medidas con el fin de describir a cada señal. La definición de algunas de estas medidas se muestra a continuación [4].

1) **Media:** Con esta medida se busca indicar con precisión el centro de los coeficientes de los componentes principales y la mejor base hallados con la DWT y la WPT respectivamente (ver anexo B). Esta medida se define con la ecuación 1.1.

2) **Varianza:** La varianza es la suma de las desviaciones de la media elevadas al cuadrado, es una característica que sirve para cuantificar el grado de esparcimiento de los datos respecto a la media. Se calcula por medio de la siguiente ecuación.

$$V = \sum_{i=1}^n (x_i - m)^2 \quad (1.15)$$

Donde x es una señal y n su número de muestras.

3) **Descomposición en valores singulares (SVD):** La descomposición en valores singulares es usada porque ésta captura la información esencial de los componentes principales y la mejor base seleccionada con DWT y WPT respectivamente (ver anexo B).

Los valores singulares $s_1 \geq s_2 \geq \dots \geq 0$ (números reales no negativos) de una matriz A son raíces cuadradas de los valores propios del producto $A^T A$, y éstos se encuentran en una matriz diagonal S .

Si s_1 es el mayor valor singular se cumple que:

- ✓ Para cualquier matriz A , $s_1(A) \geq 0$ y $s_1(A) = 0$, sí y solo sí, $A=0$.
- ✓ Para cualquier matriz A y cualquier número k $s_1(kA) = |k|s_1(A)$
- ✓ Para dos matrices A y B $s_1(A+B) \leq s_1(A) + s_1(B)$

4) Energía: La energía es la representación cuadrática de los componentes principales y la mejor base (Anexo B) de la señal y está dada por la siguiente fórmula:

$$e = \sum_{i=1}^n |x_i|^2 \quad (1.16)$$

Donde x es una señal y n su número de muestras.

Los vectores de características pueden ser formados a partir de un solo tipo de análisis (temporal, frecuencial ó tiempo – frecuencia), sin embargo también pueden ser híbridos lo que implica tener más información acerca de la señal posibilitando mejorar los resultados obtenidos en el proceso de clasificación.

1.2 REDUCCIÓN DE DIMENSIONALIDAD

Uno de los problemas que se debe resolver antes de la clasificación es la dimensión del vector de características. En una señal, el número de características puede ser muy grande, debido a esto, surge la necesidad de reducir la dimensionalidad de los datos de entrada, buscando por un lado la optimización del tiempo de cálculo, evitar la redundancia al tratar con características altamente correlacionadas, y por otro lado, eliminar ciertos datos que no representan información significativa. La reducción de características también contribuye en la disminución de la complejidad del proceso de clasificación y el incremento del desempeño del clasificador. Entre los métodos de reducción de dimensionalidad, se encuentran los de reducción por selección de características y los de reducción por proyección de características [5], [7].

1.2.1 Reducción por selección de características

El método de reducción por selección de características consiste en reducir el número de variables por la selección del mejor subconjunto de características del conjunto de la señal original, de acuerdo a algunos criterios como la separabilidad de clases [7]. La selección de características puede estar formada por:

- Un procedimiento sistemático que puede ser encontrado para la búsqueda de un subconjunto de características.
- Un criterio que puede ser establecido, para juzgar si un subconjunto de características es mejor que otro.

1.2.2 Reducción por proyección de características

El método de proyección de características consiste en determinar la mejor combinación de las características iniciales, para así formar un conjunto de características de menor dimensionalidad. Esta técnica es útil y efectiva en reconocimiento de patrones, pues reduce la complejidad de las características resultantes y además reduce la dependencia lineal entre los datos. Uno de los métodos de proyección de características, el cual se puede utilizar para que la representación de una señal sea más precisa, es el PCA (Análisis de Componentes Principales), éste busca conocer qué tan exacta se puede representar una señal con un conjunto de características, es decir, si un pequeño conjunto de características puede ser usado para reconstruir la señal exactamente, entonces se podría decir que las características son efectivas y son útiles para representar esta señal [7].

El PCA es un algoritmo lineal que basa su funcionamiento en la correlación entre variables. El análisis PCA busca unas componentes principales, sobre las que se pueden proyectar las contribuciones de cada una de las variables; proyectando las

variables sobre las dos primeras componentes principales se puede obtener información sobre la relación entre estas variables. La cercanía entre variables suele presuponer una buena correlación entre ellas, esto significa que entregan información similar, por el contrario, si las variables están distantes unas de otras indica que están fuertemente decorrelacionadas, existiendo disimilitud en la información entregada por ellas [5].

1.3 CLASIFICACIÓN DE PATRONES

La fase de clasificación es un componente importante de los sistemas inteligentes y puede considerarse como la tarea final de procesado, pues una vez definido el experimento, recogidas y pre-procesadas las señales convenientemente, y definido el vector de características representativo, el último paso es la implementación de un clasificador que pueda decidir a qué clase corresponde cada una de las señales, y de este modo generar una nueva señal para el control de los dispositivos de salida (prótesis).

La entrada al clasificador es el conjunto de características (patrón de entrada) extraídas en el paso anterior (extracción de características, con reducción de dimensionalidad si es necesario), y la salida es una indicación (índice) de la pertenencia de un objeto ó señal a una u otra clase (figura 1).

Se define un *patrón* como una descripción estructural o cuantitativa de un objeto, mediante un conjunto de descriptores o características. Su representación puede ser vectorial, es decir, patrones consistentes en un número finito de medidas (los descriptores), organizadas en un vector de dimensión finita que se denomina *vector de características* (1.17).

$$x = (x_1, x_2, \dots, x_N) \quad (1.17)$$

Donde x_i es el i -ésimo descriptor o característica y N es el número de descriptores.

Se define una *clase de patrones* como una familia de patrones que comparten algunas propiedades comunes. Las clases se representan por $C = (c_1, c_2, \dots, c_M)$, donde M es el número de clases.

El clasificador, también llamado discriminador de clases, es el encargado de asignar clases (índices) a dichos patrones de entrada y para conseguir este propósito divide el espacio de entrada en regiones de decisión, en donde idealmente, todos los patrones de una misma clase se agruparán en la misma zona del espacio de medida sin que exista una superposición entre las distintas agrupaciones formadas por otras clases.

Actualmente existen un gran número de técnicas o algoritmos de clasificación o reconocimiento de patrones. Algunos de estos procedimientos de clasificación fueron desarrollados hace ya algunos años y como es lógico, algunos han sido refinados para mejorar su desempeño, mientras otros han sido desechados por mostrarse poco útiles. Entre las técnicas más comunes utilizadas en proceso de clasificación están las técnicas tradicionales como los modelos ocultos de Markov, los árboles de decisión, los algoritmos de las k-medias; los métodos basados en distancias como la distancia Euclidea y distancia Mahalanobis; los métodos estadísticos, de los cuales hacen parte los clasificadores bayesianos simples (*naiveBayes*) y los discriminadores lineales. Entre las técnicas relativamente más recientes están los basados en técnicas inteligentes como las redes neuronales (perceptrón multicapa, la red neuronal probabilística, redes ART, entre otras), máquinas de vectores de soporte (SVM), sistemas Fuzzy, y las técnicas híbridas en las que es posible combinar algunos de los métodos ya mencionados, como por ejemplo las redes Neuro - Fuzzy.

Estos algoritmos pueden ser clasificados de diferentes formas atendiendo a sus características: se pueden distinguir entre algoritmos supervisados y no supervisados si nos fijamos en los procesos de aprendizaje; lineales o no lineales según las operaciones que realicen; paramétricos o no paramétricos en función de si se hacen suposiciones iniciales sobre el proceso a modelar [12].

1.3.1 Formas de clasificación

➤ ***Aprendizaje supervisado y no supervisado***

Los métodos no supervisados se refieren a situaciones donde el objetivo es: a partir de una serie de observaciones, establecer la existencia de clases o grupos en los datos (*clusters*). En este caso no existe un conocimiento a priori de qué clases existen, ni se dispone de un conjunto de objetos u observaciones etiquetadas, por lo que no se podrán utilizar para diseñar las funciones discriminantes. En el caso de la clasificación supervisada, las clases están predefinidas y se dispone de patrones previamente clasificados. El objetivo consiste básicamente en establecer una regla, (proceso de entrenamiento), para ubicar nuevas observaciones en alguna de las clases existentes [13].

➤ ***Métodos lineales y no lineales***

Se dice que un clasificador es lineal cuando es posible expresar su frontera de decisión en términos de una función lineal [12] (figura 2.a). En el caso de los clasificadores no lineales esta función puede ser arbitraria (figura 2.b) ó realizar el proceso de clasificación de las muestras mediante hiperplanos en un espacio de mayor dimensión.

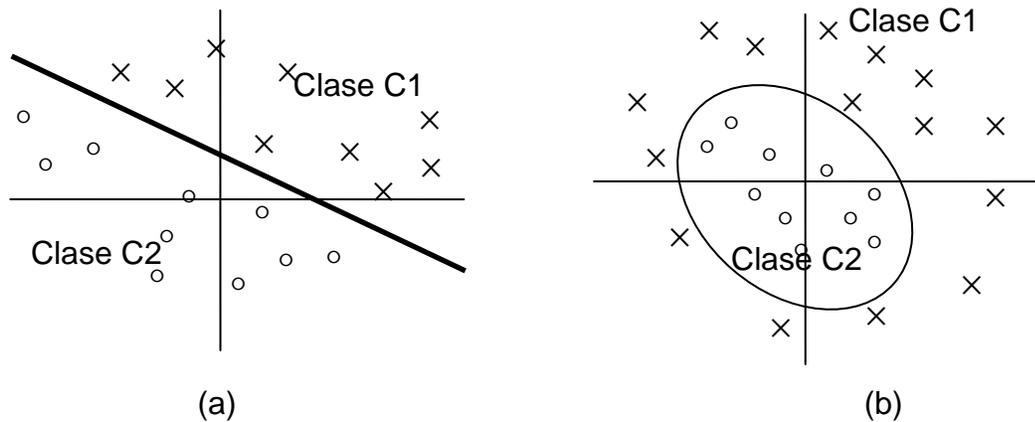


Figura 2. a) Problema separable linealmente, b) Problema no separable linealmente

➤ **Métodos de aprendizaje paramétricos y no paramétricos**

La suposición básica en métodos de aprendizaje paramétricos, es que los factores desconocidos son parámetros que involucran la densidad de probabilidad. De esta manera, en el aprendizaje a partir de muestras, el objetivo es encontrar los valores adecuados de estos parámetros [14]. Los métodos paramétricos son fáciles y eficientes, pero si la suposición sobre los valores de los parámetros es incorrecta se pueden obtener resultados erróneos.

Los métodos no paramétricos son métodos de aprendizaje en los cuales el conocimiento acerca de la forma funcional de la distribución de probabilidad condicional no está disponible ó no es usada explícitamente. El nombre puede sugerir que no se involucran parámetros, sin embargo, estos métodos a menudo requieren más parámetros que los métodos paramétricos. La diferencia radica en que: en los métodos no paramétricos los parámetros necesarios no son los relacionados con la distribución condicional [14].

1.3.2 Consideraciones para la evaluación de un clasificador

Algunas consideraciones generales que se pueden tener en cuenta para obtener un buen clasificador dependiendo de la aplicación son [13]:

- **Exactitud o Precisión:** Representa el nivel de confianza del clasificador, usualmente representado como la proporción de clasificaciones correctas que es capaz de producir. La precisión consiste, también, en la capacidad de dar las respuestas correctas a pesar de algunas equivocaciones o errores que pueden ocurrir durante el proceso de la detección y la adquisición de la señal. En el caso de las señales EMG, por ejemplo, estos errores generalmente son el resultado de la mala ubicación de los electrodos sobre la piel o por interferencia (ruido).

Una herramienta utilizada para evaluar la precisión y que además indica los errores y aciertos cometidos en el proceso de clasificación es la matriz de confusión, ó matriz de clasificación (figura 3). En esta matriz las filas representan la clase a la que la señal pertenece y las columnas hacen referencia a la clase en la cual dicha señal fue clasificada. De este modo, en la diagonal principal de dicha matriz quedan ubicadas la cantidad de muestras clasificadas correctamente (Ok) y en las otras posiciones las muestras clasificadas de manera incorrecta (x) así:

	Mov1	Mov2	Mov3	Mov4
Mov1	Ok	X	X	X
Mov2	X	Ok	X	X
Mov3	X	X	Ok	X
Mov4	X	X	X	Ok

Figura 3. Matriz de confusión.

- **Velocidad o rapidez:** Es el tiempo que toma hacer la clasificación. En algunas ocasiones la velocidad es tan importante como la precisión, por ejemplo, un clasificador con un 90% de precisión, sería preferible a uno con 95% de precisión, si el primero es 100 veces más rápido que el segundo.

- **Potencia de clasificación:** Es la razón entre la tasa de éxito y el tiempo empleado en el proceso de clasificación, es decir, a mayor tasa de éxito y menor tiempo, se obtiene mayor potencia del clasificador.

- **Flexibilidad:** Adaptabilidad del clasificador a un entorno cambiante. Es decir, este debe comprender correctamente los vectores de características, incluso si esas características son considerablemente diferentes para cualquier característica que ha sido vista antes. Debe ser capaz de adaptarse a señales provenientes de diferentes fuentes, sin que esto afecte su buen desempeño.

2. FUNDAMENTOS TEÓRICOS DE LOS CLASIFICADORES

La elección del clasificador viene supeditada tanto al dominio del problema a tratar, como a las consideraciones para la evaluación que tengan más relevancia según dicho dominio. Por esta razón es necesario tener en cuenta las características y la naturaleza de la información que se utiliza como datos de entrada del sistema y las condiciones ó características deseadas a la salida del mismo. En general, se requiere que la complejidad del clasificador sea pequeña, que opere de manera rápida, y posea buena capacidad de generalización, es decir, que sea capaz de hacer buenas predicciones con otros datos no presentes en el entrenamiento

Las señales EMG que serán sometidas al proceso de clasificación, fueron procesadas previamente en el proyecto de grado “Extracción de Características Descriptoras de Movimientos de la Mano a Partir de Señales Electromiográficas (EMG) Aplicando Técnicas Wavelets”, realizado por Paola Gasca Durán y Álvaro Rojas Rojas, estudiantes de la Universidad del Cauca, bajo la dirección del Ingeniero Harold Romo. Estas señales tienen definido, previamente, el tipo de movimiento de mano al que pertenecen, lo que a su vez define el número de clases existentes; condiciones que hacen posible utilizar clasificadores supervisados, para lo que es necesario, además, dividir los datos en dos grupos: datos de entrenamiento y datos de validación.

Las clases definidas son cuatro, lo que muestra que el problema de clasificación es multiclase, además los datos no son separables linealmente, por lo que es necesario utilizar preferiblemente clasificadores no lineales.

Los métodos escogidos en este proyecto para el desarrollo del clasificador de señales EMG son:

- Red neuronal multicapa (MLP) con algoritmo de aprendizaje backpropagation.
- Red neuronal probabilística (PNN).
- Máquina de vectores soporte (SVM).
- Red Fuzzy ARTMAP (FAM)

A continuación se exponen los fundamentos teóricos de las técnicas empleadas.

2.1 PERCEPTRÓN MULTICAPA (MLP)

Dentro de las redes neuronales, una de las más utilizadas para problemas de clasificación de patrones son las Perceptrón Multicapa o MLP (por sus siglas en inglés *Multi-layer Perceptrón*). La MLP es una red neuronal artificial (RNA), supervisada, no lineal y no – paramétrica, formada por múltiples capas, lo que le permite resolver problemas que no son linealmente separables (ver capítulo 1, figura 2.b). Se trata de una red unidireccional y su arquitectura típica es la mostrada en la figura 4.

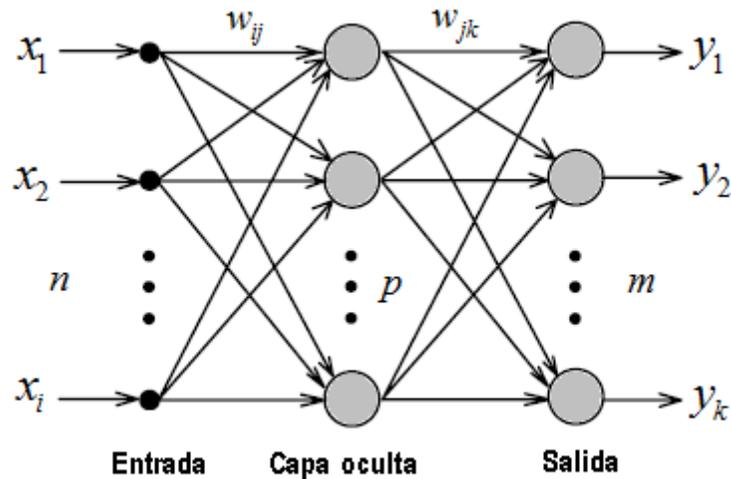


Figura 4. Arquitectura típica de una MLP

La estructura presentada en la figura 4 corresponde a una red MLP compuesta por n unidades en la capa de entrada, p unidades en la capa oculta, y m unidades

de salida. La relación entre las salidas y_k y las entradas x_i , de la red MLP es definida por (2.1).

$$y_k = \tilde{g} \left(\sum_{j=0}^p w_{jk} g \left(\sum_{i=0}^n w_{ij} x_i \right) \right) \quad (2.1)$$

Donde w_{ij} denota un peso en la capa de entrada, que va de la entrada i a la unidad j de la capa interna, y w_{jk} representa un peso en la capa interna que va de la unidad j a la salida y_k . Se ha distinguido también entre la función de activación para la capa interna g y la función de activación para la capa de salida \tilde{g} [15].

En la capa de entrada no se realiza ningún cálculo, ella sólo ejerce la función de propagar las entradas de la red a todos los elementos de la siguiente capa y por tanto no contiene unidades activas. La capa final de salida es la responsable de entregar los resultados finales y enviar la señal fuera del sistema; sus funciones de activación pueden ser lineales o sigmoideas, dependiendo del tipo de salida que se desee. El resto de capas (internas) se denominan capas ocultas, ya que no están en contacto directo con el exterior. Las unidades activas que componen la red son perceptrones con la modificación de tener como función de activación, en lugar de la función escalón, una función no constante, continua, monótona creciente y acotada, como las funciones sigmoide o la tangente hiperbólica [15], [16], [17] representadas por las ecuaciones (2.2) y (2.3) y mostradas en la figura 5a y 5b respectivamente.

Función sigmoide:

$$g(\alpha) = \frac{1}{1 + \exp(-\alpha)} \quad (2.2)$$

Función tangente hiperbólica:

$$g(\alpha) = \frac{e^{\alpha} - e^{-\alpha}}{e^{\alpha} + e^{-\alpha}} \quad (2.3)$$

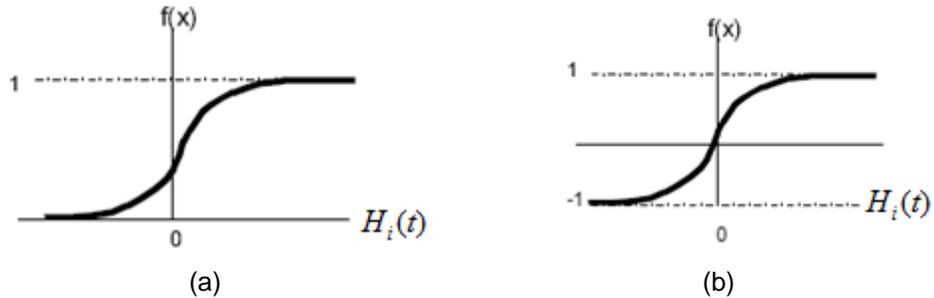


Figura 5. (a) Función logarítmica sigmoidea, (b) Función tangente sigmoidea

El número de neuronas o nodos de entrada es determinado según el número de características que contienen los vectores, y las neuronas de salida son definidas para cada aplicación en particular dependiendo del número de clases de salida y el tipo de función de activación a utilizar. Respecto al número de capas ocultas y la cantidad de neuronas que debe tener cada una de ellas no se conoce una metodología general para su determinación; aunque este tema se ha tratado en diversos trabajos, los resultados son poco concluyentes o demasiado difíciles de aplicar en la práctica. En términos generales, son determinados empíricamente para cada aplicación.

La habilidad de aprendizaje de esta red a partir del entrenamiento es muy potente, y tiene la capacidad de resolver problemas de clasificación no lineal, Sin embargo el comportamiento no lineal de las neuronas las cuales están muy interconectadas, hace que sea difícil conocer a priori la respuesta de la red, además la existencia de neuronas ocultas, impide “ver” cómo se produce el aprendizaje y determinar cuáles son las características que mejorarían el aprendizaje [16].

El algoritmo de aprendizaje más utilizado para el entrenamiento de las redes MLP es el algoritmo o regla *back propagation*. El algoritmo *back-propagation* estándar está basado en el gradiente, en el cual los pesos de la red son actualizados a lo largo de la dirección contraria al gradiente de la función de costo que mide el error cometido por la red. El término *backpropagation* se refiere al método usado para actualizar los pesos en función del error, que es una aplicación de la regla de la cadena para el cálculo de derivadas del cálculo elemental. Básicamente el entrenamiento consiste en lo siguiente [16], [17]:

- a) Pasada hacia delante (*forward pass*): Son calculadas las salidas de cada capa y el error en las unidades de salida de la red.
- b) Pasada hacia atrás (*backward pass*): El error de las salidas es utilizado para alterar los pesos de las unidades de salida. Luego mediante propagación hacia atrás del error en las unidades de salida se calcula el error en las neuronas de la capa oculta y los pesos en ella son alterados usando este error.

El algoritmo *backpropagation* padece dos defectos graves. Por un lado, el gradiente es un indicador extremadamente local de la función de cambio óptimo. De forma que, si para zonas próximas entre sí de la superficie del error, el gradiente toma direcciones opuestas, se producen fluctuaciones que provocan que el tiempo de búsqueda del mínimo sea considerablemente largo. Por otro lado, no se sabe *a priori* cuál es el tamaño del cambio de los pesos más adecuado para una tarea dada [18]. Un ritmo de aprendizaje demasiado pequeño ocasiona una disminución importante en la velocidad de convergencia y un aumento en la probabilidad de acabar atrapado en un mínimo local. En cambio, un ritmo de aprendizaje demasiado grande conduce a inestabilidades en la función de error con el peligro de poder pasar por encima del mínimo global. Por tanto, los valores de estos parámetros de aprendizaje, en muchas ocasiones, se deben determinar mediante ensayo y error [19].

La importancia de las redes MLP está en su capacidad de auto-adaptar los pesos de las neuronas para aprender la relación que existe entre el conjunto de vectores o patrones de entrada y su correspondiente salida, y poder aplicar esa relación después del entrenamiento a nuevos vectores de entrada imperfectos o con ruido. Esta capacidad se conoce como generalización. La red debe encontrar una representación interna que le permita generar las salidas deseadas durante la etapa de entrenamiento, y posteriormente durante el funcionamiento ser capaz de generar salidas para entradas que no le fueron mostradas durante el aprendizaje pero que se asemejan a alguna de las que si le fueron mostradas [17].

En la aplicación de las redes MLP para la clasificación de patrones el proceso que se ejecuta es el siguiente [19]:

Paso 1. Inicialización: todo el conjunto de pesos w_{ij} y umbrales θ_j de la red se establece de manera aleatoria.

Paso 2. Pasada hacia delante (*cálculo forward*): En esta fase se aplica en la capa los vectores de entrada, ésta propaga su efecto a través de las diferentes capas y como consecuencia produce un vector de salida. Durante este proceso, los pesos sinápticos de la red son fijos y no se modifican.

Entonces, partiendo del conjunto de entrenamiento denotado por $[x(n), y(n)]$, con el vector de entrada $x(n)$ aplicado a la capa de entrada y la respuesta deseada $y(n)$ presentada en la capa de salida. El nivel de actividad interna de la red $v_j^{(l)}(n)$ para la neurona j en la capa l es dado por:

$$v_j^{(l)}(n) = \sum_{i=0}^p w_{ji}^{(l)}(n) d_i^{(l-1)}(n) \quad (2.4)$$

Donde $d_i^{(l-1)}(n)$ es la señal para la neurona i en la capa previa $l-1$ en la iteración n y $w_{ji}^l(n)$ es el peso de la neurona j en la capa l que es conectada a la neurona i en la capa $l-1$ en la iteración n .

Para $i=0$, tenemos

$$d_0^{(l-1)} = -1 \quad (2.5)$$

y

$$w_{j0}^l(n) = \theta_j^l(n) \quad (2.6)$$

Donde $\theta_j^l(n)$ es el umbral aplicado a la neurona j en la capa l .

Con el uso de la función sigmoïdal como función de activación, la salida de la neurona j en la capa l es dada por:

$$d_j^{(l)}(n) = \frac{1}{1 + \exp(-v_j^l(n))} \quad (2.7)$$

Si la neurona j está en la primera capa oculta ($l=1$), se determina

$$d_j^{(0)}(n) = x_j(n) \quad (2.8)$$

Donde $x_j(n)$ es el j^{th} elemento del vector de entrada $x(n)$.

Si la neurona j está en la capa de salida ($l=L$), se tiene:

$$d_j^{(L)} = o_j(n) \quad (2.9)$$

Siendo $o_j(n)$ la salida obtenida.

Entonces la señal de error puede ser calculada como:

$$e_j(n) = y_j(n) - o_j(n) \quad (2.10)$$

Donde $y_j(n)$ es la salida deseada del j^{th} elemento.

Paso 3. Pasada hacia atrás (*cálculo backward*): aquí los pesos se modifican de acuerdo con la regla de corrección del error. La señal de salida real se compara con la señal deseada y como resultado se obtiene una señal de error, que se propaga en dirección contraria a través de la red modificando los pesos, de forma que, al volver a pasar el vector de entrada hacia adelante, la respuesta obtenida se asemeje más a la salida deseada.

Se calcula el gradiente local (δ) de la red en avance regresivo, capa por capa. Para la neurona j en la capa de salida L , el gradiente local es dado por:

Para la neurona j en la capa de salida L .

$$\delta_j^{(l)}(n) = e_j^{(L)}(n) o_j(n) [1 - o_j(n)] \quad (2.11)$$

Para la neurona j en la capa oculta l .

$$\delta_j^{(l)}(n) = o_j(n) [1 - o_j(n)] \sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) \quad (2.12)$$

Los pesos de la red en la capa l pueden ser actualizados de acuerdo a la regla delta generalizada así:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \alpha [w_{ji}^{(l)}(n) - w_{ji}^{(l)}(n-1)] + \eta \delta_j^{(l)}(n) d_i^{(l-1)}(n) \quad (2.13)$$

Donde η es la tasa de aprendizaje y α es la constante momentum.

Paso 4. Calcular de nuevo el error total y si todavía el error total no es satisfactorio, se regresa al paso 2 y se repite el proceso hasta obtener una tasa de error deseada.

2.2 REDES NEURONALES PROBABILÍSTICAS (PNN)

La red es utilizada en problemas de clasificación, gracias a su simplicidad, generalidad y rapidez de aprendizaje. El ajuste de pesos involucrados no es iterativo por lo que su aprendizaje es directo, lo que aumenta considerablemente la velocidad de entrenamiento. Por otra parte, a diferencia de la red MLP, las redes PNN requieren de más neuronas, pero pueden diseñarse en menos tiempo y además no presenta el problema de los óptimos locales [12], [20].

La red neuronal artificial probabilística es una familia de las redes neuronales de Base Radial y es una continuación directa del trabajo sobre clasificadores de Bayes [20], en donde la estimación de las densidades de probabilidades de cada clase se realiza utilizando funciones de base radial centradas en cada patrón de entrenamiento, como se verá a continuación.

La red PNN consta de cuatro capas, una capa de entrada que contiene m nodos, uno por cada rasgo del vector de características, una capa de neuronas de base radial o capa de patrones con un número de neuronas igual al número de vectores de entrenamiento, una capa competitiva¹ o sumatoria cuyo número de neuronas es igual al número de categorías consideradas en el problema de clasificación [12], [22], [23] y la capa de salida cuyos nodos son neuronas binarias que producen la decisión de clasificación. En la figura 6 se muestra un esquema general de la arquitectura de esta red.

¹ En las capas competitivas las neuronas de salida compiten entre sí para ser las únicas activas. De este modo la neurona de salida que da el valor más alto se le asigna el valor total de 1 (activa), mientras que a las demás se le da un valor de 0 (inactivas).

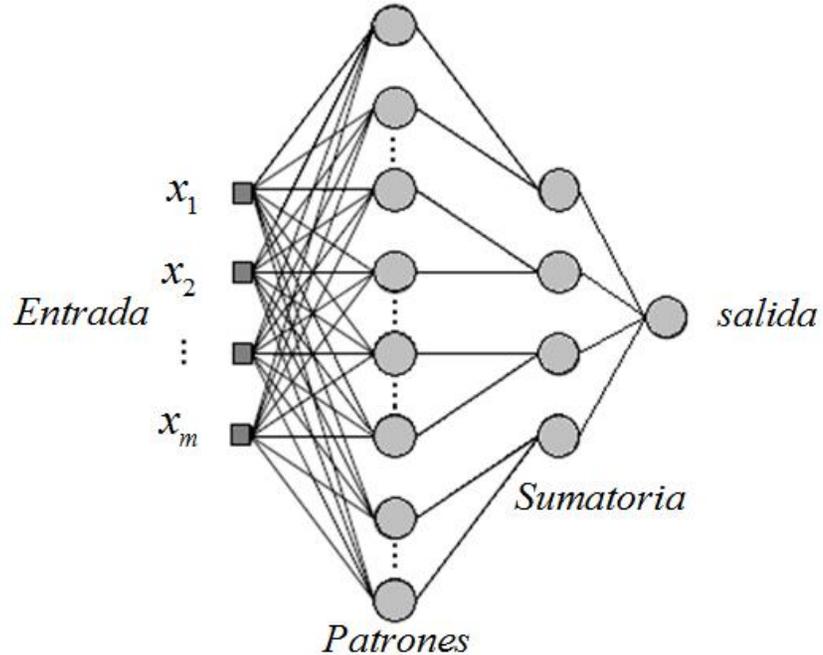


Figura 6. Esquema básico de una PNN.

Durante el entrenamiento de la red, se presenta un vector de entrada $X = (x_1, x_2, \dots, x_m)$ a las neuronas de entrada que cumplen el rol de distribución, es decir, envían cada nodo correspondiente a un rasgo del vector de características, a todos los nodos de la segunda capa, de esta forma, todos los nodos en la capa de patrones reciben el vector de características X completo.

Las neuronas de la capa de patrones son divididas en k grupos, uno por cada clase que se va a reconocer, después se aplica el algoritmo de k -medias sobre el conjunto de entrada, para calcular los centros² de las Funciones de Base Radial (RBF) y almacenar estos valores como pesos w [24].

² Existe un centro para cada función radial involucrada en la capa oculta. Regularmente, definen un vector de la misma dimensión del vector de entrada y hay normalmente un centro diferente por cada nodo de la capa oculta.

En la fase de clasificación o reconocimiento, cuando se presenta un nuevo vector de entrada, en la segunda capa se calculan las distancias Euclídeas (d) entre el vector de entrada (x) y los pesos de cada neurona (w), para obtener en cada neurona de base radial un escalar (también llamada contribución) indicativo del parecido entre el vector a clasificar y los pesos de dicha neurona [12], [20], [21], [22].

$$d = |x - w| = \sqrt{(x_1 - w_1)^2 + (x_2 - w_2)^2 + \dots + (x_m - w_m)^2} \quad (2.14)$$

Si para una neurona el escalar es cero, entonces el vector a clasificar ha resultado ser idéntico al vector de pesos de dicha neurona [12].

Los escalares resultantes (d) del cálculo de la distancia son multiplicados por un escalar denominado “*spread*” que es idéntico para todas las neuronas de la capa de base radial de la red e introducido como argumento en una función de tipo Gaussiana denominada “*radbas*” [12].

La expresión de la función “*radbas*” es:

$$radbas(n) = e^{-n} \quad (2.15)$$

En donde

$$n = \sum_{i=1}^k \frac{(d_i)^2}{2\sigma^2} \quad (2.16)$$

En la ecuación (2.16) k corresponde al número de muestras, i a la muestra i -ésima y finalmente el parámetro σ es denominado “*spread*”, y su función es determinar la anchura de la campana de Gauss [12].

Si se escoge un valor para el “*spread*” suficientemente bajo la red se comporta como el clasificador k -NN (k - *nearest neighbour*), pues los nuevos vectores deben ser muy parecidos a los establecidos como pesos por lo que responde sólo a un

tipo de clasificación y la categorización puede fallar. Por el contrario si este valor se hace muy grande la red diseñada tendrá en consideración varios vectores cercanos y no será tan exacta, pues aunque los nuevos vectores sean muy diferentes pueden ser clasificados dentro de una misma clase [12], [20] y de esta manera generar errores.

La figura 7 muestra el aspecto de dicha función que adquiere su valor máximo (igual a la unidad) cuando su argumento vale cero. Por lo tanto, dado un vector nuevo que deba ser clasificado, la neurona de base radial que produzca un máximo a su salida será aquella cuyos pesos se parezcan más al vector de entrada [12].

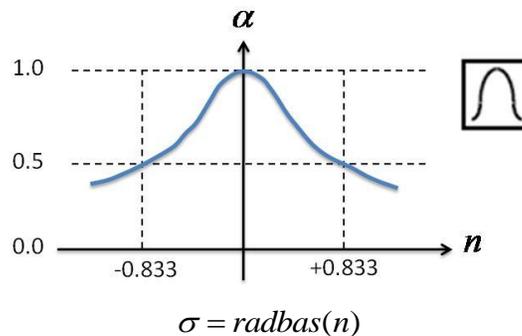


Figura 7. Función de transferencia de una neurona de base radial

La capa de sumatorias es la encargada de sumar las contribuciones de las diferentes neuronas de la capa de base radial, para generar un vector de probabilidades de dimensión igual al número de categorías existentes. Por ejemplo, cada neurona en la capa de la suma recibe todas las salidas de la capa de patrones asociados con una clase dada. Si diez neuronas en la capa de patrones calcularon la función exponencial para la clase 1 las diez salidas de la capa de patrones serían sumadas por una sola neurona de la capa de suma y se obtiene un vector de probabilidades que pasa a la capa de salida en donde se utiliza función de transferencia “*competitiva*” que escoge el máximo entre todas las probabilidades, y produce un 1 para esa clase y un 0 para las demás [12], [20].

En otras palabras, cuando resulta ganadora la neurona de base radial i esto implica que el vector de entrada pertenece a la categoría j . Entonces los pesos del enlace entre la neurona de base radial i y la neurona competitiva j es igual a la unidad mientras que los pesos que enlazan a la neurona i con las demás neuronas competitivas (las otras categorías) son cero. El elemento de la posición i -ésima del vector resultante de sumar las contribuciones de las neuronas de base radial puede ser interpretado como la probabilidad de que el vector de entrada pertenezca a la categoría i -ésima [12].

La “función” de activación en la red PNN, consiste en la densidad de probabilidades (DP) entre la entrada y los patrones de entrenamiento. Como se ve, no hay un entrenamiento previo para cálculo de pesos, sino que la capa de patrones calcula el DP para cada vector de entrada, comparándolo con los vectores de entrenamiento para las clases o categorías de salida pre-especificadas [20].

2.3 MAQUINA DE VECTORES DE SOPORTE (SVM)

Las máquinas de vectores de soporte, conocidas bajo las siglas SVM (Support Vector Machines), son una familia de algoritmos de clasificación y regresión desarrollados por Vladimir Vapnik y colaboradores a principios de los años 90 [25]. Desde entonces las SVM han ganado un merecido reconocimiento gracias a los sólidos principios teóricos en los que se fundamenta su diseño, su alta capacidad de generalización y uso de funciones núcleo o kernel. Las SVM, son un método de clasificación supervisado, no paramétrico y aplicable a problemas de separación de clases lineal y no lineal. La arquitectura de las SVM sólo depende de un parámetro de penalización denotado como C y la función kernel (incluyendo sus parámetros). De esta manera se evitan requerimientos sobre parámetros exclusivos de arquitectura, tales como número de nodos, capas y el tipo de conexión entre capas, como es el caso de las redes neuronales [25].

2.3.1 Clasificación utilizando SVM

El planteamiento original de las SVM se centró en resolver problemas de clasificación binaria, donde el objetivo es generar un modelo capaz de separar los objetos de dos clases distintas por medio de un hiperplano óptimo, y en donde la solución puede ser escrita por la combinación de unos pocos puntos de entrada que son los llamados vectores de soporte. Para los casos en que las clases son linealmente separables el hiperplano se calcula en el espacio original. Sin embargo, en general esta linealidad no existe; en tales casos, las SVM aplican una transformación del espacio de entrada a un espacio multidimensional en el cual las clases si son linealmente separables. Esta transformación no lineal se realiza de distintas maneras, empleando mapeos, también llamados kernels, tales como los Polinomiales y RBF. Bajo esta transformación es posible separar las clases en el espacio de salida con un hiperplano [26].

Las SVM implementan algoritmos de clasificación con aprendizaje supervisado y se asume que se parte con n datos de entrenamiento N dimensionales (\vec{x}_i) . Siendo x el vector de características que representa al objeto i con su respectiva etiqueta (y_i) , tal como se muestra en (2.17):

$$x_i \in \mathfrak{R}^N \quad \text{y} \quad y_i \in \{+1, -1\} \quad (2.17)$$

Se busca estimar una función f tal que para una entrada \mathfrak{R}^N produzca una salida en $\{\pm 1\}$, según se muestra en (2.18):

$$f : \mathfrak{R}^N \rightarrow \{+1, -1\} \quad (2.18)$$

2.3.2 Análisis del caso linealmente separable

➤ Hiperplano clasificador óptimo

Los clasificadores con SVM se fundamentan en la obtención de hiperplanos que separen los datos de entrenamiento en dos subgrupos. Entre cada una de las clases, etiquetadas como $\{+1, -1\}$, existe un único hiperplano óptimo de separación (HOS). Se busca que la distancia entre el hiperplano óptimo y el patrón de entrenamiento más cercano sea máxima, con la intención de forzar la generalización de la máquina de aprendizaje [27].

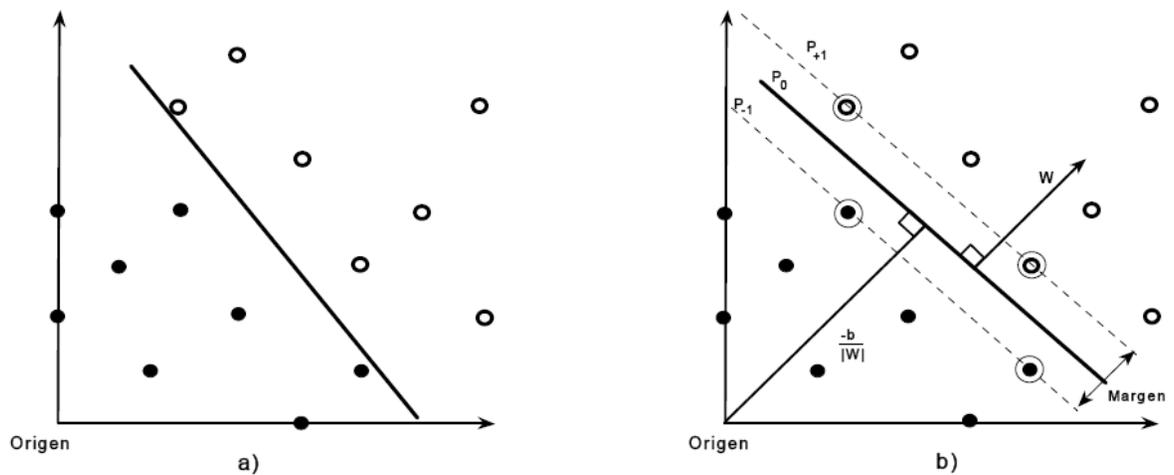


Figura 8. Hiperplanos que separan correctamente un conjunto de datos.
a) Hiperplano de separación de datos. b) HOS con mayor margen de separación entre clases. Los datos encerrados con círculos son denominados vectores de soporte (VS).

El HOS de la figura 8, se presenta en (2.19)³:

$$g(\vec{x}) = \langle \vec{w} \cdot \vec{x} \rangle + b = 0 \quad (2.19)$$

³ Un plano N dimensional de la forma $a_n x_n + a_{n-1} x_{n-1} + \dots + a_j x_j + a_0$, se puede reescribir de la forma $\langle \vec{w} \cdot \vec{x} \rangle + b = 0$. $\vec{w} = [a_n, a_{n-1}, \dots, a_1]$ es un vector perpendicular al hiperplano, $\vec{x} = [x_n, x_{n-1}, \dots, x_1]$, y $b = a_0$

Para maximizar el margen, se proponen dos planos paralelos que contienen los puntos más cercanos al HOS. Si $1/\|\vec{w}\|$ es la distancia entre los puntos más cercanos y el plano HOS, las ecuaciones de los planos están dadas por (2.20) y (2.21)⁴:

$$P_{+1} = \langle \vec{w} \cdot \vec{x} \rangle + b = +1 \quad (2.20)$$

$$P_{-1} = \langle \vec{w} \cdot \vec{x} \rangle + b = -1 \quad (2.21)$$

El margen definido como la distancia perpendicular entre (2.20) y (2.21) está dado por (2.22)

$$\begin{aligned} [\langle \vec{w} \cdot \vec{x}_{+1} \rangle + b] - [\langle \vec{w} \cdot \vec{x}_{-1} \rangle + b] &= +1 - (-1) \\ \vec{w} \cdot (\vec{x}_{+1} - \vec{x}_{-1}) &= 2 \\ \frac{\vec{w}}{\|\vec{w}\|} \cdot (\vec{x}_{+1} - \vec{x}_{-1}) &= \frac{2}{\|\vec{w}\|} \end{aligned} \quad (2.22)$$

La distancia entre el HOS y el origen de coordenadas está dada por (2.23)

$$\text{Si } \langle \vec{w} \cdot \vec{x}_i \rangle + b = 0 \text{ y } \vec{x}_i = 0 \Rightarrow \frac{\langle \vec{w} \cdot \vec{x}_i \rangle + b}{\|\vec{w}\|} = \frac{b}{\|\vec{w}\|} \quad (2.23)$$

Por definición, entre los dos hiperplanos no deben existir datos de entrenamiento y por tanto los datos deben cumplir con (2.24) y (2.25).

$$\langle \vec{w} \cdot \vec{x} \rangle + b \geq +1 \quad \text{para } y_i = +1 \quad (2.24)$$

$$\langle \vec{w} \cdot \vec{x} \rangle + b \leq -1 \quad \text{para } y_i = -1 \quad (2.25)$$

Luego la función de decisión $f_{w,b}(\vec{x}_i) = y_i$, corresponde al signo que resulta de evaluar la función $g(\vec{x})$ en la ecuación del HOS (2.19) con un dato \vec{x}_i , tal como se muestra en (2.26)

⁴ La distancia perpendicular que hay entre cualquier punto a un plano es $(\langle \vec{w} \cdot \vec{x}_i \rangle + b) / \|\vec{w}\|$.

$$f_{w,b}(\bar{x}_i) = \text{sign}[g(\bar{x}_i)] = \text{sign}[\langle \bar{w} \cdot \bar{x}_i \rangle + b] \quad (2.26)$$

Combinando (2.24) y (2.25) se obtiene (2.27):

$$y_i(\langle \bar{w} \cdot \bar{x}_i \rangle + b) \geq 1 \quad (2.27)$$

Si existe un hiperplano que satisfaga (2.27), se dice que los datos son linealmente separables. Para encontrar el HOS se debe maximizar el margen (2.22), teniendo en cuenta la restricción (2.27), lo que es equivalente a resolver el problema planteado en (2.28) y (2.29).

$$\min_{w,b} \frac{1}{2} \langle \bar{w} \cdot \bar{w} \rangle = \min_{w,b} \frac{1}{2} \|\bar{w}\|^2 \quad (2.28)$$

$$\text{Sujeto a } y_i(\langle \bar{w} \cdot \bar{x}_i \rangle + b) \geq 1, \forall i \quad (2.29)$$

La función (2.28) se llama función objetivo, y junto con (2.29) se conforma un problema de optimización cuadrático con restricciones. Los problemas de este tipo se tratan introduciendo el método de los multiplicadores de Lagrange.

➤ **Solución al problema de optimización con restricciones**

Para las restricciones de la forma $R_i \geq 0$, cada restricción se multiplica por α_i (un multiplicador de Lagrange positivo), y se restan de la función objetivo (2.28), para así formar la función de Lagrange presentada en (2.30) [27].

$$L(\bar{w}, \bar{\alpha}, b) = \frac{1}{2} \|\bar{w}\|^2 - \sum_{i=1}^n \alpha_i [y_i(\langle \bar{w} \cdot \bar{x}_i \rangle + b) - 1] \quad (2.30)$$

La función de Lagrange (2.30) se debe minimizar con respecto a las variables primarias w y b , y maximizada sobre los α_i para encontrar los planos sobre los

que se encuentran los vectores de soporte (en la figura 8 P_{+1} y P_{-1}). Para el caso de $y_i(\langle \bar{w} \cdot \bar{x}_i \rangle + b) - 1 > 0$, el correspondiente α_i debe ser cero, debido a que éste es el valor que maximiza a (2.30). Los α_i diferentes de cero son para el caso en que $y_i(\langle \bar{w} \cdot \bar{x}_i \rangle + b) - 1 = 0$, que corresponden a los patrones de entrenamiento que quedan sobre los hiperplanos paralelos al HOS dados por las ecuaciones (2.20) y (2.21). Este último enunciado corresponde a las condiciones de Karush – Kuhn – Tucker o condiciones complementarias de optimalidad, presentadas de (2.31) y (2.33) [28].

$$\alpha_i [y_i(\langle \bar{w} \cdot \bar{x}_i \rangle + b) - 1] = 0, \forall i \quad (2.31)$$

$$\frac{\partial}{\partial w} L(\bar{w}, \bar{\alpha}_i, b) = 0 \Rightarrow \bar{w} = \sum_{i=1}^n \alpha_i y_i \bar{x}_i \quad (2.32)$$

$$\frac{\partial}{\partial b} L(\bar{w}, \bar{\alpha}_i, b) = 0 \Rightarrow \sum_{i=1}^n \alpha_i y_i = 0 \quad (2.33)$$

La solución de \bar{w} en (2.32) queda en función de un subconjunto de patrones de entrenamiento cuyo multiplicador de Lagrange es diferente de cero. Es decir, el soporte de \bar{w} está en los patrones de entrenamiento más cercanos al HOS. De aquí el nombre de Máquinas de Vectores Soporte.

Reemplazando (2.32) en (2.30) y teniendo en cuenta que se cumpla (2.33), se eliminan las variables primarias \bar{w} y b llegando así al problema dual de optimización, el cual se resuelve mediante (2.34):

$$\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle \bar{x}_i \cdot \bar{x}_j \rangle \quad (2.34)$$

$$\text{Sujeto a las restricciones } \alpha_i \geq 0, \forall i \text{ y } \sum_{j=1}^n \alpha_j y_j = 0 \quad (2.35)$$

Siendo $\alpha_i \geq 0 \quad \forall i$ una propiedad de los α_i , cuando se introducen los multiplicadores de Lagrange, y $\sum_{j=1}^n \alpha_j y_j = 0$ el resultado cuando se diferencia el Lagrangiano (2.30) con respecto a b .

Este problema puede ser resuelto con métodos de programación cuadrática estándar. [27] Una vez obtenido el vector $\vec{\alpha}_i$, se puede obtener \vec{w} y b :

$$\vec{w} = \sum_{i=1}^{VS} \alpha_i y_i \vec{x}_i \quad (2.36)$$

$$b = -\frac{1}{2} (\langle \vec{w}, x_r \rangle + x_s) \quad (2.37)$$

Donde x_r y x_s son los vectores soporte de cada una de las clases que satisfacen $\alpha_r \alpha_s > 0$.

Ahora, la ecuación del HOS (2.19) y la función de decisión se pueden expresar como se presentan en (2.38) y (2.39).

$$g(\vec{x}) = \sum_{i=1}^{VS} [\alpha_i y_i \langle \vec{x}_i \cdot \vec{x} \rangle] + b \quad (2.38)$$

$$f(\vec{x}) = \text{sign}(\sum_{i=1}^{VS} [\alpha_i y_i \langle \vec{x}_i \cdot \vec{x} \rangle] + b) \quad (2.39)$$

La utilización de los multiplicadores de Lagrange se debe a dos razones fundamentales:

- Las restricciones (2.29) están en función de α_i (2.35), que las hace más fáciles de resolver.
- En la reformulación del problema, los datos de entrenamiento \vec{x}_i sólo aparecen en forma de productos punto entre ellos mismos (2.34), (2.38) y (2.39).

➤ **Características de la solución**

- Las SVM reducen al mínimo la dimensión del modelo ya que solo depende de los datos con mayor información como los son los vectores de soporte, esto hace que tengan una excelente habilidad para la generalización y sean menos propensas a sufrir sobre-entrenamiento (*Over-Fitting*) [12].
- w es una combinación lineal de un pequeño número de datos.
- Muchos de los α_i son cero, ya que no todos los datos estarán sobre las fronteras de decisión.
- Los vectores x_i con α_i distintos de cero son llamados los Vectores de Soporte (VS). Las fronteras de decisión están determinadas solo por los VS.
- Para clasificar un nuevo dato Z no es necesario formar w explícitamente, simplemente se calcula:

$$w \cdot z + b = \sum_{i=1}^{VS} \alpha_i y_i \langle \vec{x}_i \cdot \vec{z} \rangle + b$$

Siendo $t_i (i=1, \dots, VS)$ los índices de los vectores de soporte y donde existe una separación lineal de las clases.

2.3.3 Solución al caso linealmente no separable con variables de relajación

La implementación de clasificadores basados en SVMs desarrolladas como se presenta en las secciones anteriores, puede tener altos errores, debido a que en la práctica es poco probable que existan datos linealmente separables, y si existe, no siempre es la mejor solución para el problema de clasificación ya que pueden existir datos erróneos, ruido o alto traslapamiento de clases en los datos de entrenamiento y ésta solución puede no ser la mejor.

Debido a lo anterior, se propone otra alternativa que busque el mejor hiperplano tolerando ruido en los datos de entrenamiento. Cortes y Vapnik proponen una solución, basándose en el documento [29], para permitir la posibilidad de ejemplos que violen la restricción (2.29), considerando variables de relajación (ξ), como se presenta en la figura 9, en donde se puede ver que hay un dato en la parte equivocada del plano de separación [30]

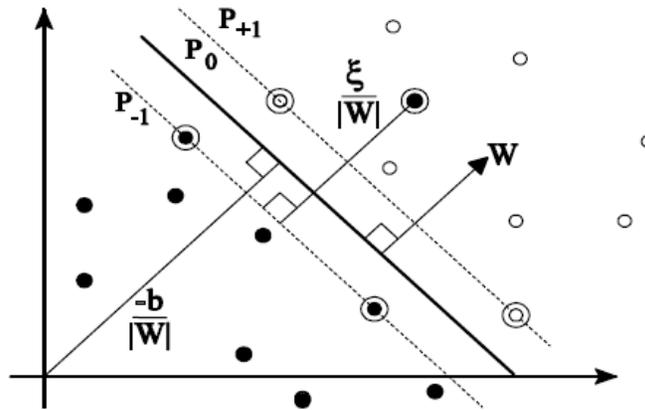


Figura 9. Hiperplano lineal clasificador para el caso no separable.

La nueva restricción se presenta en (2.40):

$$y_i(\langle \bar{w} \cdot \bar{x}_i \rangle + b) \geq 1 - \xi_i, \forall i \quad (2.40)$$

Donde:

$$\xi_i \geq 0, \forall i \quad (2.41)$$

Dadas estas nuevas restricciones, a continuación se encuentra el clasificador que mejor generaliza para obtener mejores resultados con datos linealmente no separables.

Una posible forma de obtener el hiperplano óptimo con margen débil es minimizando la función (2.42) sujeta a (2.43).

$$\min_{w,b} \frac{1}{2} \langle \bar{w} \cdot \bar{w} \rangle + C \sum_{i=1}^n \xi_i \quad (2.42)$$

$$\text{Sujeto a: } y_i (\langle \bar{w} \cdot \bar{x}_i \rangle + b) \geq 1 - \xi_i, \forall i \quad (2.43)$$

Un alto valor del parámetro C corresponde a una alta penalización a los errores. Con los multiplicadores de Lagrange el problema se transforma en (2.44), sujeto a (2.45):

$$\max_{\alpha} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle \bar{x}_i \cdot \bar{x}_j \rangle \quad (2.44)$$

$$\text{Sujeto a: } 0 \leq \alpha_i \leq C, \forall i \text{ y } \sum_{i=1}^n \alpha_i y_i = 0 \quad (2.45)$$

El hiperplano separador solución se puede expresar como (2.38), y la función de decisión como (2.39) [31].

2.3.4 Máquinas de vectores soporte no lineal

En la mayoría de los casos reales los datos no son linealmente separables debido a que se encuentran vectores de una clase dentro de la región correspondiente a vectores de otra clase y, por consiguiente, nunca podrían ser separados con el método lineal. Para abordar este problema es necesario desarrollar una nueva estrategia, asunto que se describe a continuación [12].

El principio de las SVM no lineales consiste en mapear o establecer una relación entre el espacio de entrada y un nuevo espacio de representación de dimensión alta, a través de una función Φ no lineal elegida a priori, esto con el fin de lograr que las clases en el nuevo espacio sean linealmente separables tal como se presenta en la figura 4 [26].

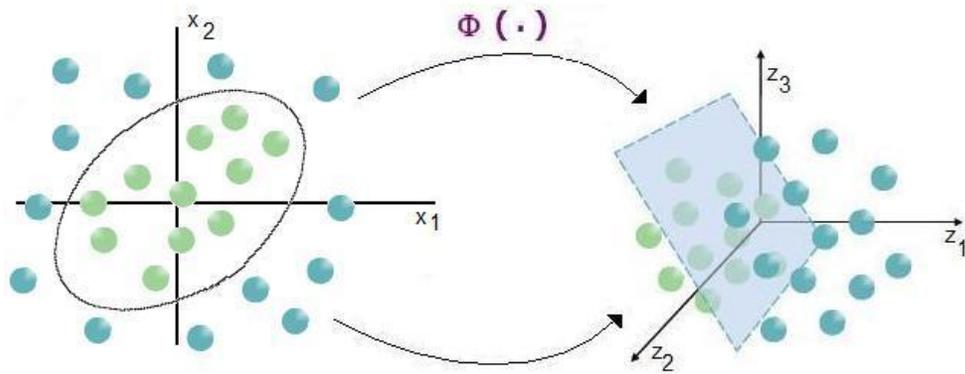


Figura 10. SVM no lineal representada en una nueva dimensión alta.

Por medio de una función (Φ) se trazan los datos de entrada $(\vec{x}_i \in \mathfrak{R}^N)$ a algún espacio de mayor dimensión y con producto punto definido (2.46). Este espacio se llama espacio característico (F) .

$$\Phi: \mathfrak{R}^N \rightarrow F \quad (2.46)$$

Así, la función (2.38) que depende del producto punto de los vectores en el espacio de entrada, pasa a una función que depende del producto punto de los vectores en el espacio característico, como se muestra en (2.47).

$$g(\vec{x}) = \sum_{i=1}^n [\alpha_i y_i \langle \Phi(\vec{x}_i) \cdot \Phi(\vec{x}) \rangle] + b \quad (2.47)$$

Entonces se define una nueva función, que sea el producto punto de los vectores en el espacio característico, llamada función kernel (k) presentada en (2.48):

$$k \langle \vec{x}_i, \vec{x} \rangle = \langle \Phi(\vec{x}_i) \cdot \Phi(\vec{x}) \rangle \quad (2.48)$$

Debido a que F es de dimensión alta, el lado derecho de la ecuación (2.48) es costoso en términos computacionales. Sin embargo existe una función kernel (k) , que puede evaluarse eficazmente y se puede demostrar que corresponde a un trazado de (Φ) en un espacio que abarca todos los productos punto.

Con la función kernel no se necesita definir explícitamente la función Φ , ya que ésta entrega directamente el resultado del producto punto, que es lo que realmente interesa para la aplicación de clasificación. Generalizando, se puede probar que de cada función kernel que presente una matriz definida positiva, se puede construir una función Φ que cumpla con (2.48). Los kernel utilizados en este trabajo son el polinomial (2.49) y función de base radial (RBF) (2.50) [31].

$$k(\vec{x}, \vec{y}) = (\vec{x} \cdot \vec{y} + c)^d \quad \text{para } c > 0 \quad (2.49)$$

$$k(\vec{x}, \vec{y}) = e^{-\frac{\|\vec{x}-\vec{y}\|^2}{2\sigma^2}} \quad (2.50)$$

Al incluir la función kernel se puede reescribir (2.44) sujeta a (2.45), como (2.51):

$$\underset{\alpha}{\text{máx}} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j k(\vec{x}_i \cdot \vec{x}_j) \quad (2.51)$$

Sujeto a las restricciones: $0 \leq \alpha_i \leq C, \forall i$ y $\sum_{i=1}^n \alpha_i y_i = 0$

Las ecuaciones del HOS (2.38) y función de decisión (2.39) se reescriben como (2.52) y (2.53).

$$g(\vec{x}) = \sum_{i=1}^n [\alpha_i y_i k(\vec{x}_i \cdot \vec{x})] + b \quad (2.52)$$

$$f(\vec{x}) = \text{sign} \left(\sum_{i=1}^n [\alpha_i y_i k(\vec{x}_i \cdot \vec{x})] + b \right) \quad (2.53)$$

2.3.5 SVM multiclase

El planteamiento anterior sólo aborda el problema de la biclasificación (clases con valores de ± 1), pero muchos problemas son de más de dos clases ($y_i \in \{1, 2, \dots, l\}, l > 2$). Para resolver el problema de multclasificación con máquinas de vectores soporte se admiten dos tipos de arquitecturas, las multclasificadoras y las biclasificadoras generalizadas [32]. En este trabajo se estudiarán solamente las biclasificadoras.

➤ **Máquinas biclasificadoras generalizadas**

Este tipo de máquina construye una función clasificadora global a partir de un conjunto de funciones biclasificadoras. Existen técnicas de descomposición y reconstrucción que permiten a las SVM biclasificadoras manejar problemas de multclasificación con mayor simplicidad y/o menor tiempo de respuesta que una SVM generaliza a multclasificación.

a. Arquitecturas de descomposición

En el esquema de descomposición estándar se construyen m máquinas biclasificadoras en paralelo, que son entrenadas con sus respectivos datos de entrenamiento. Para esto se crea una matriz de descomposición, donde los elementos de una clase son asignados a salidas positivas, los de otras a salidas negativas. Existen dos técnicas de descomposición que son “uno contra el resto” y “uno contra uno” de las cuales se estudiará esta última por ser la más rápida debido a que trabaja con menos muestras que la técnica “uno contra todos”.

- **Uno contra uno (1 –vs– 1)**

Se realiza implementando $m = l(l-1)/2$ clasificadores binarios.

El entrenamiento de la i -ésima SVM se realiza con solo 2 de las l clases existentes ($l > 2$) en el grupo de n datos de entrenamiento. Se asigna la etiqueta positiva ($t_j = +1$) a los datos n_j que pertenecen al subgrupo de datos de la clase j donde $j \in \{1, \dots, l\}$, y la etiqueta negativa ($t_p = -1$) a los datos (n_p) que pertenecen al subgrupo de datos de la clase p donde $p \in \{1, \dots, l\}$ y $p \neq j$. Los

demás datos ($n_r = n - n_j - n_p$) no se utilizan en el entrenamiento de la i -ésima SVM y por lo tanto son etiquetados con cero ($t_r = 0$), creándose una matriz de descomposición (D_{1-v-1}) presentada en (2.54).

$$D_{i,j} = \begin{cases} +1 & \text{si } n_h \in n_j \\ -1 & \text{si } n_h \in n_p \\ 0 & \text{si } n_h \in n_r \end{cases} \quad (2.54)$$

b. Método de reconstrucción

Cada máquina biclasificadora entrenada emite una respuesta en forma numérica $Z_i = g_i(\vec{x})$ a una entrada (\vec{x}). La información más importante en esta respuesta, en principio, se encuentra en el signo $S^i = f_i(\vec{x}) = \text{sign}(g_i(\vec{x}))$ que adopta la función de decisión.

➤ **Esquemas de votación**

Se realiza un proceso de votos donde el número máximo que puede tener una determinada clase es $\ell - 1$ ($\ell =$ número de clases), esta es la forma de reconstrucción más habitual y consideran sólo el signo de las predicciones de todas las máquinas biclasificadoras. Estos signos se interpretan en función de las clases implicadas en las máquinas biclasificadoras utilizado en el esquema de descomposición. Para la votación de la i -ésima máquina biclasificadora:

$$\theta(S^i) = \begin{cases} y_i & \text{si } S^i = +1 \\ 0 & \text{si } S^i = -1 \end{cases} \quad (2.55)$$

Tras la interpretación de las predicciones, el elemento de combinación realiza un recuento del número de clases votadas, acción de la que toma el nombre de esquema de reconstrucción, que posee diferentes variantes como lo son: votación

por unanimidad, votación por mayoría absoluta y votación por mayoría simple, siendo éste último el esquema que se va a trabajar en este documento y consiste en determinar como respuesta final, aquella única clase que haya obtenido más votos que el resto de clases.

2.4 REDES FUZZY ARTMAP

La red Fuzzy ARTMAP⁵ es una red neuronal que, basada en la Teoría de la Resonancia Adaptativa (ART), es capaz de aprender a clasificar patrones analógicos de una forma supervisada. Esta red, es una modificación de la red ARTMAP⁶, en donde se ha cambiado el módulo ART1⁷ por el Fuzzy-ART⁸ [34], razón por la cual es necesario exponer, aunque de manera breve, los fundamentos de las otras arquitecturas de red, a partir de las cuales surge la Fuzzy ARTMAP. La exposición de las redes ART, ART-1 y ARTMAP se encuentran en el anexo C y la de las redes Fuzzy - ART y Fuzzy ARTMAP se hace a continuación.

2.4.1 Fuzzy ART

Fuzzy ART es una evolución del algoritmo ART-1, diseñado para el agrupamiento de patrones de entrada analógicos con aprendizaje no supervisado. La arquitectura Fuzzy ART se muestra en la figura 11. Tiene la misma estructura que el sistema ART1 (anexo C). Consiste en dos capas de neuronas, F1 y F2, y un subsistema de vigilancia controlado por el parámetro $\rho \in [0,1]$, en donde ρ recibe el nombre de “factor de vigilancia” [34].

⁵ Fuzzy ARTMAP: Arquitectura de clasificación basado en ART con entrenamiento supervisado (MAPPING) para patrones de entrada analógicos.

⁶ ARTMAP: Arquitectura de clasificación basada en ART con entrenamiento supervisado (MAPPING) para patrones de entrada binarios.

⁷ ART1: Arquitectura de agrupamiento con entrenamiento no supervisado para patrones de entrada binarios.

⁸ Fuzzy-ART: Arquitectura de agrupamiento con entrenamiento no supervisado para patrones de entrada analógicos.

La capa F1 es la capa de entrada y está formada por N neuronas. Cada neurona de entrada recibe una componente x_i , entre 0 y 1, de un vector de entrada X . La capa F2 es la capa de categorías o clases y está compuesta por M neuronas que reciben entradas T_j . Todas las neuronas de ambas capas están completamente interconectadas entre sí, pero ahora, a diferencia de la ART1, los pesos que conectan las capas en dirección de arriba a abajo y de abajo a arriba son analógicos e iguales entre sí ($z_{ij} = z_{ij}^{bu} = z_{ij}^{td}$) [34]. El superíndice bu viene de las siglas en inglés *bottom-up*, que significa de abajo a arriba, e indica la dirección de la conexión (de la capa F1 a la F2), El superíndice td viene de las palabras *top-down*, y quiere decir de arriba a abajo.

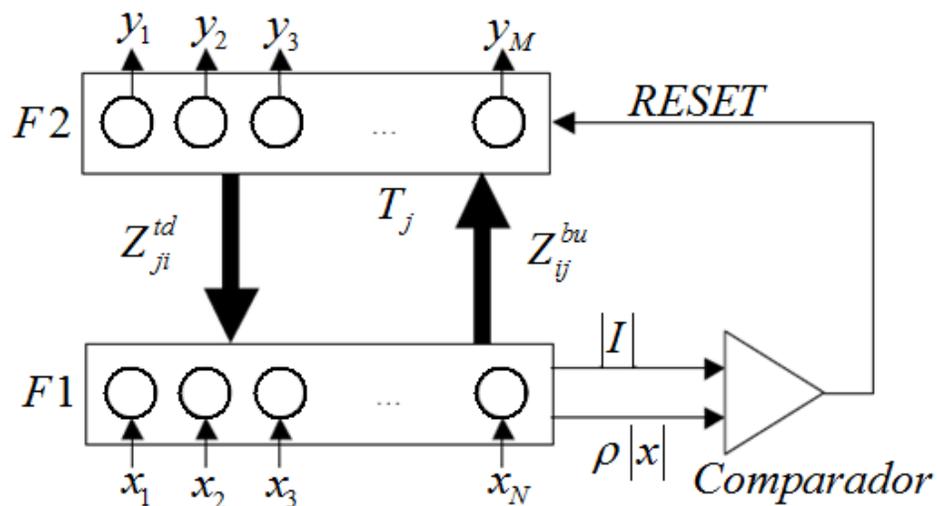


Figura 11. Esquema de la red Fuzzy ART

Su modo de funcionamiento consiste en que cada vez que la red recibe un nuevo vector de entrada X reacciona activando uno y solo uno de los nodos de salida (y_1, y_2, \dots, y_M). Cada uno de estos nodos representa una de las diferentes clases que se han creado con las entradas anteriores. En caso de que la medida no se parezca lo suficiente a ninguno de los nodos ya asignados se creará uno nuevo que representará una nueva clase cuyo primer miembro será este vector [12].

La entrada a una neurona j de la capa F2 viene dada por:

$$T_j = \sum_{i=1}^N z_{ij}^{bu} x_i, \quad j = 1, \dots, M \quad (2.56)$$

Los términos T_j se denominan *funciones de elección* y representan una cierta distancia entre el patrón almacenado $z_j^{bu} = (z_{1j}^{bu}, z_{2j}^{bu}, \dots, z_{Nj}^{bu})$ y el patrón de entrada $x = (x_1, x_2, \dots, x_N)$.

Las neuronas de la capa F2 tienen sus salidas en "0", excepto la neurona que reciba la mayor T_j . A esta neurona se le denota con el subíndice J , y tiene salida "1" [34].

$$y_J = 1, \quad \text{Si } T_J = \max_j \{T_j\} \quad (2.57)$$

$$y_{j \neq J} = 0$$

La entrada a una neurona i de F1 es:

$$V_i = \sum_{j=1}^M z_{ji}^{td} y_j = z_{ji}^{td}, \quad i = 1, \dots, N \quad (2.58)$$

En la figura 11 se observa un comparador, que representa al *subsistema de vigilancia*, el cual se encarga de comprobar lo que se ajusta del patrón de entrada a la categoría activa en F2. Este subsistema de vigilancia cuenta con dos parámetros, el parámetro de vigilancia o "factor de vigilancia" (ρ) y el ritmo de aprendizaje (β). El factor de vigilancia determina lo riguroso que debe ser el algoritmo a la hora de agrupar medidas, si éste parámetro es muy cercano a la unidad implica una clasificación muy exigente, de manera que dos medidas deben ser muy parecidas para ser agrupadas en una misma clase, por el contrario, un parámetro cercano a cero permite la agrupación de medidas poco parecidas, lo

que genera como resultado una red con pocos nodos de salida, ya que el número de clases diferentes es reducido. Por su parte, el ritmo de aprendizaje queda controlado por el parámetro (β) , siendo su valor igual a la unidad para un aprendizaje rápido e igual a cero en caso de que no se deba aprender más [33].

Las principales diferencias entre las arquitecturas ART1 y Fuzzy-ART son:

- 1) Los vectores de entrada son analógicos para la Fuzzy-ART.
- 2) En Fuzzy-ART sólo hay un conjunto de vectores de pesos analógicos z_j .
- 3) En el cálculo de las funciones T_j , de la regla de aprendizaje, y del criterio de vigilancia, la intersección (\cap , AND binario) se sustituye por el operador borroso MIN (\wedge , AND analógico).

En redes ART con entradas analógicas, como Fuzzy-ART y Fuzzy-ARTMAP, para evitar el problema de proliferación de categorías cuando la norma de los vectores de entrada x es demasiado pequeña, se normalizan⁹ dichos vectores dividiéndolos por su norma. Pero esto a menudo no es conveniente, porque se pierde la información sobre la amplitud de los patrones de entrada, razón por la que se suele utilizar la regla de codificación complementaria para solucionar el problema, ya que ésta preserva la información de amplitud a la vez que normaliza los vectores de entrada [34].

La *codificación complementaria*, asegura el buen funcionamiento del algoritmo y se define como el complemento de x como $x^c = 1 - x$. Si x es un vector binario (x_1, x_2, \dots, x_N) , la codificación complementaria consiste en duplicar el número de componentes del patrón de entrada. En lugar de utilizar x , se usa el vector binario $2N$ -dimensional [6], [34].

⁹ El normalizado es una técnica que consiste en dividir todos los valores de cada patrón para lograr que el valor máximo esté acotado ó que los patrones de la señal no superen la unidad.

$$(x, x^c) = (x_1, x_2, \dots, x_N, x_1^c, x_2^c, \dots, x_N^c) \quad (2.59)$$

Con $x_i^c = 1 - x_i$

El hecho de que la norma de los patrones de entrada x sea la misma para todos, evita la proliferación de clases o categorías. En redes ARTMAP con patrones de entrada binarios, si las normas de las entradas son iguales, se evita que el algoritmo pueda no llegar a funcionar.

2.4.2 Red Fuzzy ARTMAP

La red Fuzzy ARTMAP es una generalización a vectores analógicos (con componente entre cero y uno) de la red binaria ARTMAP (ver anexo C) [12], [33], [34], en la que, además, el operador intersección (\cap) de ARTMAP se ha sustituido por el operador mínimo borroso (\wedge) de la lógica Fuzzy. Es un método no lineal y no paramétrico que está formada por dos redes fuzzy ART (figura 11) conectadas entre sí, por el vector de relaciones denominado “memoria asociativa” o “*mapfield*” (figura 12) [12], [33], [34].

En la fase de entrenamiento, a la red denominada Fuzzy Art-A (figura 12) le llegan los vectores de entrada (X) y a la red Fuzzy Art-B le llegan los vectores que codifican la categoría correcta de cada medida del conjunto de entrenamiento (Y) [6], [34].

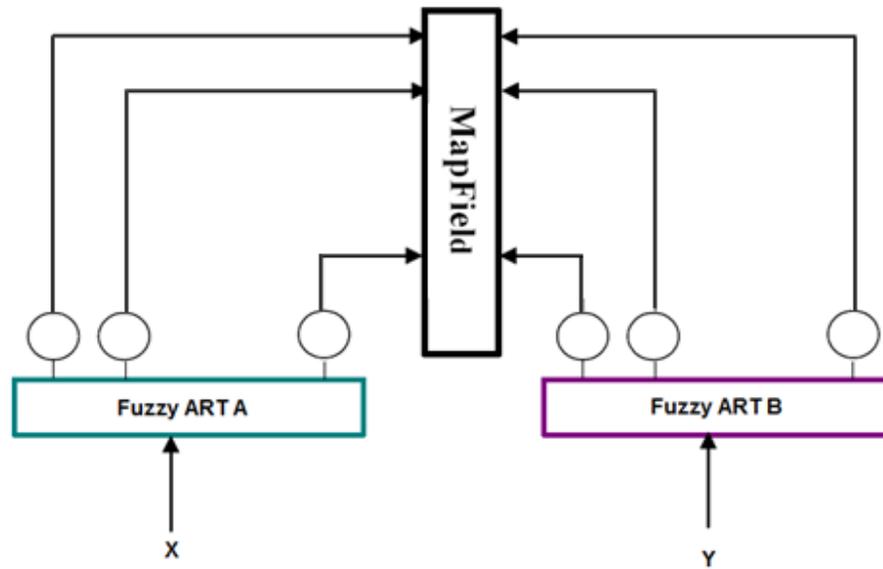


Figura 12. Esquema general de una red Fuzzy ARTMAP

Inicialmente en la red A el factor de vigilancia es cero. En la red B se suele dar un valor igual a la unidad, ya que las medidas que deban ser clasificadas conjuntamente enviarán a la red B codificaciones idénticas. Además cualquier vector de codificación diferente, por parecido que sea el resto, debe ser detectado y debe activar una neurona de salida diferente en la red B [12], [33].

Cada vez que se suministra una medida de entrenamiento, la red A activa un nodo y la B activa otro. La memoria asociativa que las une aprende a relacionar nodos activados. De esta forma a cada nuevo nodo que se activa en la red A se le asocia un nodo B. Cabe destacar que, normalmente, cada uno de los nodos en B serán imagen de varios nodos A (cada categoría contiene varias medidas) mientras que cada una de las categorías creadas en A sólo tendrá una imagen en B (cada medida solo puede pertenecer a una categoría) [12], [33].

Cuando una nueva medida activa un nodo en A ya existente, se comprueba si la imagen de ese nodo asignada por la memoria asociativa coincide con el nodo que se ha activado en B paralelamente. En el caso de que no coincidan, se incrementa

el valor del parámetro de vigilancia hasta que la neurona que se active en A tenga por imagen alguna neurona activada en B. En definitiva, el valor de vigilancia en A solo se incrementa lo estrictamente necesario para que la red A separe en nodos diferentes las medidas que deben estar clasificadas en diferentes categorías [12], [33].

Al igual que en Fuzzy ART, en Fuzzy ARTMAP los parámetros utilizados son el factor de vigilancia (ρ) y el ritmo de aprendizaje (β).

En su implementación práctica, se presenta un problema que debe ser tratado con sumo cuidado y consiste en que la red aprende el 100% de las medidas del conjunto de entrenamiento sacrificando lo mínimo posible la generalización. Sin embargo, la presencia de medidas erróneas en el conjunto de entrenamiento puede requerir un incremento del valor de vigilancia excesivo, lo cual perjudicará seriamente la capacidad de generalizar de la red [12].

Esta arquitectura presenta una serie de propiedades de las que también gozan las redes Fuzzy-ART y ARTMAP. Entre las más interesantes se encuentran [34]:

- 1) Aprendizaje estable. Esto se debe al hecho de que los pesos, ya sean del módulo Fuzzy-ART como del *Map-Field*, sólo pueden decrecer con el tiempo.
- 2) Número pequeño de parámetros. Los parámetros de diseño son ρ y β . Además el sistema funcionará independientemente de los valores que se den a estos parámetros, siempre que se encuentren en los rangos especificados.
- 3) La arquitectura y los valores iniciales son siempre los mismos, independientemente del problema que se esté resolviendo.
- 4) El subsistema que controla el valor del parámetro de vigilancia crea un número mínimo de categorías Fuzzy-ART necesarias para cumplir con el criterio de vigilancia. Lleva a cabo una regla de aprendizaje *minimax* que permite a la red

aprender rápido, eficientemente y con precisión, a la vez que minimiza el error de predicción.

- 5) Al igual que en la red Fuzzy-ART, existe una relación entre el parámetro de vigilancia y el tamaño de las categorías establecidas. Cuando menor sea ρ , mayor será la capacidad de generalización y la compresión de código. Por el contrario, cuando mayor sea ρ , menor tamaño tiene cada categoría Fuzzy-ART y menor error de predicción, aunque existirán más pesos y nodos de categorías.

3. CLASIFICADORES IMPLEMENTADOS.

En este capítulo se presenta cómo están conformados los vectores de características que se utilizan como entrada de los clasificadores implementados y luego de hacer una descripción general de éstos, se describe con cierto detalle la implementación de cada clasificador en particular.

Con el objeto de poder tener acceso a los clasificadores y a las bases de datos de los seis pacientes diferentes (ver apartado 3.1) con las características DT-DWT o DT-WPT, ya sea para entrenamiento o clasificación, se desarrolló un programa en Matlab 7.0, con interfaz grafica llamado “Clasificación de señales EMG” (ver figura 17), en la que es posible observar la matriz de confusión (ver capítulo 1, figura 3), el porcentaje de acierto y de error, como también el tiempo de procesamiento de la CPU para el proceso de clasificación. Para conocer más acerca del programa ver el anexo D.



Figura 13. Interfaz grafica del programa CLASIFICACION DE SEÑALES EMG.

3.1 DESCRIPCIÓN DE LA BASE DE DATOS

Los vectores de características disponibles para el proceso de clasificación se obtuvieron a partir de la extracción de características de las señales EMG originadas por seis (6) pacientes diferentes [5]. Para obtener dichas señales se ubicaron 16 sensores en diferentes partes del antebrazo, luego los pacientes ejecutaron en dos tiempos diferentes cuatro (4) movimientos de mano que en orden son: reposo (M1), agarre fino (M2), agarre grueso (M3), mano abierta (M4). Los registros de las señales EMG originales están almacenadas en dos matrices de 256x160, una para cada canal, correspondientes a 20 segundos de grabación de cuatro (4) movimientos, a 1024 muestras por segundo (5 segundos de grabación por cada movimiento) [5]. Estas señales fueron estudiadas y analizadas en el proyecto de grado “Extracción de Características Descriptoras de Movimientos de La Mano a Partir de Señales Electromiográficas (Emg) Aplicando Técnicas Wavelets” desarrollado por los estudiantes de Ingeniería Electrónica y Telecomunicaciones, Diana Paola Gasca Duran y Luis Alvaro Rojas Rojas, bajo la dirección del Ingeniero Harold Armando Romo Romero [5], en donde se seleccionaron solo dos canales (para cada uno de los pacientes), correspondientes a las señales que generaron un menor índice de separabilidad de clases (CSI)¹⁰, lo que representa una mejor caracterización de las señales.

La caracterización se logro utilizando las características individuales de cada movimiento, las cuales se obtuvieron empleando las transformadas DWT y WPT (ver anexo B), además de un análisis en el dominio del tiempo (DT), tal como se describe más adelante en el apartado 3.1.1.

¹⁰ Este índice proporciona el grado de traslape entre las diferentes clases. Por lo tanto, entre más pequeño sea su valor, mayor será la separabilidad que existe entre las clases, lo cual significa que la caracterización es mejor.

El procedimiento de reducción de dimensionalidad también fue considerado en el trabajo de grado mencionado, pues éste ayuda a mejorar el desempeño y la eficiencia de los algoritmos de clasificación, ya que los vectores de características contienen mayor cantidad de información relevante y la menor dimensión posible [5]. Esta fase fue implementada aplicando Análisis de Componentes Principales (PCA), y se evaluó el índice de separabilidad entre clases utilizando los vectores de características de dimensión reducida resultantes.

De los dos registros de las señales EMG se obtuvieron dos grupos de vectores de características almacenados en dos bases de datos diferentes, una asignada para entrenamiento y la otra para validación.

Cada base de datos, de entrenamiento y validación, se almacena en una matriz de tres (3) dimensiones ($a \times b \times c$) para cada paciente, en donde (a) representa el número de movimientos, (b) es el número de muestras y (c) el número de características. De esta forma, las matrices obtenidas tienen una dimensión de $(4 \times 40 \times 12)$, como se muestra en la Figura 14.



Figura 14. Matriz de vectores de características en 3D.

3.1.1 Descripción de los vectores de características

Las características extraídas con WPT y DWT son la media, la varianza, los valores singulares y la energía, y las del DT son la media y la longitud de onda, todas obtenidas por medio de dos canales (ver apartado 1.1).

Los vectores de características para usar en la entrada del clasificador, pueden ser vectores simples o vectores híbridos, es decir, pueden ser formados a partir de características extraídas de las señales EMG utilizando solo una técnica como WPT, DWT ó DT, ó pueden estar compuestos por características extraídas con dos o más técnicas, por ejemplo WPT – DT ó DWT – DT.

En este trabajo serán utilizados vectores híbridos, pues al tener más información acerca de la señal es posible que sean establecidas mayores diferencias entre las muestras de una y otra clase y de esta manera mejorar los resultados obtenidos en el proceso de clasificación [1]. Estos vectores están formados por cuatro (4) características DT y ocho (8) WPT ó DWT, para un total de doce (12) características para cada muestra, organizadas como se muestra en la figura 15.

1	Media –Canal 1
2	Longitud de Onda – Canal 1
3	Media –Canal 2
4	Longitud de Onda – Canal 2
5	Media –Canal 1
6	Varianza – Canal 1
7	Valores Singulares – Canal 1
8	Energía – Canal 1
9	Media –Canal 2
10	Varianza – Canal 2
11	Valores Singulares – Canal 2
12	Energía – Canal 2

■ Características DT
■ Características DWT o WPT

Figura 15. Conformación del vector de características para una muestra.

3.1.2 Clases de salida.

Los métodos de clasificación, en este documento, tienen como objetivo determinar a cuál de los cuatro movimientos básicos de mano pertenece una determinada señal EMG. Esto se logra a la salida del clasificador, donde se obtiene un índice de pertenencia de la señal EMG a una clase determinada. Las clases definidas son (figura 16):

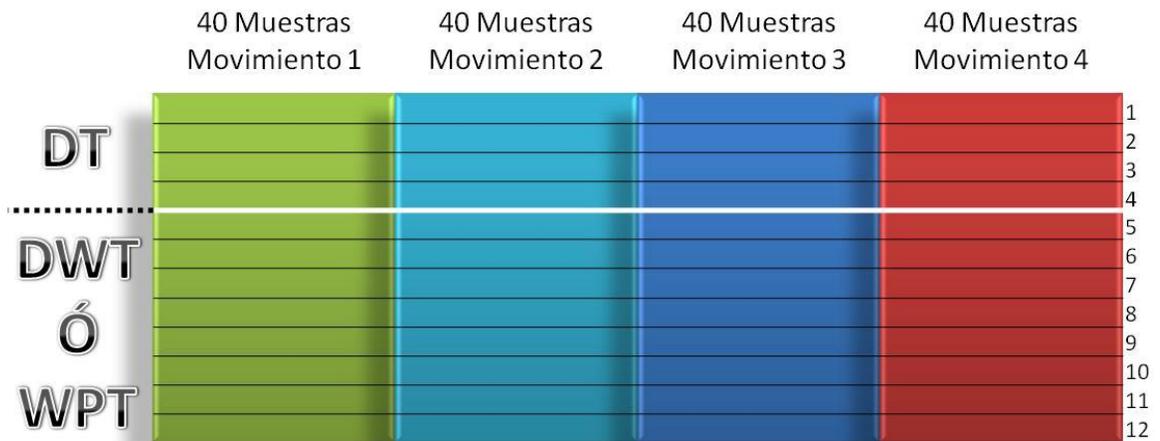
Clase 1: Mano en reposo; Clase 2: Agarre fino; Clase 3: Agarre grueso; Clase 4: Mano abierta.



Figura 16. Tipos de movimientos a clasificar.

3.1.3 Vectores de entrenamiento

Para el entrenamiento de los clasificadores se utiliza la primera base de datos, es decir, el grupo de vectores obtenidos a partir de la extracción de características de las señales EMG emitidas durante la primera ejecución de los movimientos realizada por cada paciente. Estas son 40 muestras de cada movimiento en cada uno de los registros de los pacientes, para un total de 160 muestras por paciente (figura 17).



- | | | |
|-------------------------------|---------------------------------|----------------------------------|
| 1: Media (canal 1) | 5: Media (canal 1) | 9: Media (canal 1) |
| 2: Longitud de Onda (Canal 1) | 6: Varianza (canal 1) | 10: Varianza (canal 1) |
| 3: Media (canal 2) | 7: Valores singulares (canal 1) | 11: Valores singulares (canal 1) |
| 4: Longitud de Onda (Canal 2) | 8: Energía (canal 1) | 12: Energía (canal 1) |

Figura 17. Matriz de vectores de características en 2D para la entrada del clasificador.

3.1.4 Construcción de los clasificadores.

Para la construcción de los clasificadores propuestos en este documento, primero se debe definir cuáles serán los datos de entrenamiento y organizarlos en una matriz bidimensional ($a \times b$), en donde a corresponde al número de características y es igual a 12 y b corresponde al número de muestras y es igual a 160. Como los clasificadores son de aprendizaje supervisado es necesario definir un vector $1 \times b$ para etiquetar la clase de movimiento a la cual pertenece cada muestra de entrenamiento.

Dependiendo del clasificador, se requiere definir algunos parámetros ó características para después proceder a ejecutar la creación y el entrenamiento de cada clasificador y almacenarlo. Esto se hace con el propósito de ahorrar tiempo en el proceso de clasificación de nuevos datos, pues así, es posible clasificar sin necesidad de realizar el entrenamiento cada vez que se utilice.

3.1.5 Vectores de validación

Para la validación de los clasificadores, se utiliza la segunda base de datos obtenida de la misma forma que la utilizada para el entrenamiento. De igual manera el número total de datos para validación son 160 muestras en total por cada paciente.

No sobra aclarar que las muestras usadas para el entrenamiento no fueron usados para la validación y viceversa.

3.1.6 Diagrama en bloques.

El proceso de clasificación supervisada que se utiliza en este trabajo, se puede resumir en la figura 18. En esta figura se muestra la interacción entre los cinco componentes que son: datos de entrenamiento, los nuevos datos o datos de validación, el constructor de clasificadores, el clasificador y la clase o salida.

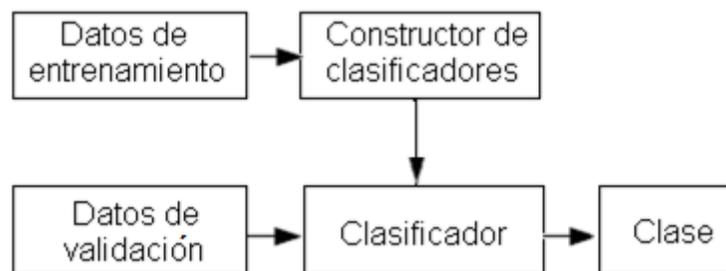


Figura 18. Esquema del proceso de clasificación

Una vez realizado el proceso de entrenamiento, ya es posible utilizar el nuevo clasificador para la categorización de los nuevos datos, que al igual que los datos de entrenamiento, son organizados en una matriz $a \times b$ ($a = 12$ características y $b = 160$ muestras). A la salida del clasificador se obtiene un vector de resultados

$1 \times b$ en el que se almacenan los índices de las clases en las que fueron clasificadas las nuevas muestras.

Con ayuda de este vector de resultados se construye la matriz de clasificación o matriz de confusión (capítulo 1) para evaluar el desempeño del clasificador en cuanto a la precisión.

3.2 CLASIFICADOR CON LA RED PERCEPTRÓN MULTICAPA (MLP)

Como fue explicado en los fundamentos teóricos (apartado 2.1), la topología de la MLP debe contener algunas capas ocultas para solucionar problemas que no son separables linealmente. En teoría, la red puede tener cualquier número de capas ocultas, sin embargo a medida que aumenta la cantidad de ellas el entrenamiento tiende a hacerse más lento y el área del espacio de pesos más complicado, lo que aumenta el coste computacional sin presentar mejoras notables en los resultados. En cuanto al número de neuronas en las capas ocultas, en general debe ser lo suficientemente grande como para que se forme una región compleja que sea capaz de resolver el problema; sin ser muy grande pues la estimación de los pesos puede ser poco confiable para el conjunto de los patrones de entrada disponibles. Por lo tanto, en este trabajo se implementó una red de 2 capas ocultas y se probó un número variable de neuronas en ellas, buscando siempre que el error obtenido fuese el mínimo posible con la topología de red más simple.

Después de probar varias configuraciones de red, finalmente se optó por la red de dos (2) capas ocultas con seis (6) neuronas en cada una de ellas y una sola neurona en la capa de salida. Como función de transferencia en las capas ocultas se utilizó la función tangente-sigmoide (*tansig*). En la capa de salida se utilizó la función de transferencia lineal (*purelin*) que permite incluir una sola neurona capaz de identificar las cuatro clases definidas.

Definidas las características de la red MLP, se procede a entrenarla. Para el entrenamiento de las redes MLP han sido desarrollados diferentes variaciones del algoritmo *backpropagation*, tales como el gradiente descendente y el gradiente descendente con momentum, métodos que en general, suelen ser demasiado lentos, por lo que en este trabajo fueron descartados. Con el propósito de mejorar la velocidad del proceso de entrenamiento, existen otros algoritmos de alta ejecución, los cuales pueden converger desde diez hasta cien veces más rápido que los de gradiente descendente. Estos algoritmos rápidos usan técnicas de optimización numérica estándar y entre ellos cabe mencionar el gradiente conjugado, quasi-Newton y el método de Levenberg – Marquardt [35].

El algoritmo Levenberg - Marquardt (LM) es una variante del algoritmo *backpropagation* (BP) y se caracteriza por su convergencia más rápida con respecto al BP, su robustez y por presentar los mejores resultados en la clasificación. El algoritmo LM, es un método de gradiente de segundo orden, es una técnica iterativa que localiza el mínimo de una función multivariable que es expresada como la suma de cuadrados de las funciones de los reales estimados no lineales. Ésta se ha convertido en una técnica estándar para problemas de mínimos cuadrados no lineales y puede ser considerada como una combinación del gradiente descendente y el método de Gauss – Newton. Cuando la solución actual está lejos de la correcta, el algoritmo actúa de la misma manera que un método de gradiente descendente: lento, pero garantizando la convergencia. Cuando la solución actual está cerca a la solución correcta, se convierte en un método de Gauss - Newton [35].

Para el entrenamiento de la red neuronal fue utilizado el algoritmo Levenberg – Marquardt, los pesos fueron inicializados aleatoriamente y el tiempo de entrenamiento para todos los pacientes fue de 300 epoch, esto porque durante la ejecución del algoritmo se vio que era un número suficiente para alcanzar el error máximo permisible de $10 - e^{-10}$. La red entrenada solo es almacenada si al ser

probada con los datos de validación, el error generado por el clasificador y verificado en la matriz de confusión, no sobrepasa un valor máximo permisible entre el 5% y el 15% según el paciente, lo que demuestra que los pesos generados si son apropiados para la posterior clasificación de nuevos patrones. El valor del error es verificado en la matriz de confusión

Después se procede a definir las nuevas muestras para validación y clasificarlas utilizando la red ya entrenada, con lo que se obtiene el vector de resultados para crear la matriz de clasificación y calcular la tasa de acierto y de error.

3.3 CLASIFICADOR CON LA RED NEURONAL PROBABILÍSTICA (PNN)

Otro de los clasificadores implementados es una red neuronal probabilística que se caracteriza por su estructura simple y la rapidez de su algoritmo de aprendizaje.

En este clasificador a diferencia del MLP, para definir la clase de cada una de las muestras se utiliza una matriz $c \times b$, en donde c es el número total de clases definidas (4 clases) y b el número de muestras (160 muestras). Esta matriz está compuesta solamente de ceros y unos, así que 1 indica la pertenencia de una muestra a una clase determinada y 0 indica la no pertenencia.

En primer lugar, las PNN utilizan el conjunto de patrones de entrenamiento para construir funciones de distribución de probabilidades de cada clase, que posteriormente son utilizadas para estimar la probabilidad de que un nuevo ejemplar, no usado para construir la red y representado por un vector de características, pertenezca a alguna de dichas clases.

Durante el entrenamiento de la red, se presentan los vectores de entrada en una matriz 12x160 (12 características y 160 muestras) a las neuronas de entrada para después pasar a representarse en la capa de patrones. La neurona

correspondiente al patrón i en el grupo k de la capa de patrones, calcula su salida usando una función gaussiana (ecuación 2.15 capítulo 2), donde σ (*spread*) determina el ancho del área en el espacio de entrada en el cual la neurona responde, así se encarga de caracterizar las distancias entre cada una de las posibles configuraciones almacenadas en los datos de entrada. Este parámetro se debe ajustar para encontrar el modelo de PNN óptimo y para hacerlo se realizaron variaciones de “*spread*” en un intervalo de 0.5 a 3.5, se entrenó la red, se evaluó y repitió el proceso hasta que se encontró la mejor clasificación. Para seleccionar “*spread*” fueron tenidas en cuenta las consideraciones expuestas en el apartado 2.2 del capítulo 2.

En definitiva, la arquitectura de la red quedó conformada por 12 neuronas en la capa de entrada (número de características), 160 en la capa de patrones (número de vectores de entrenamiento), 4 neuronas en la capa de sumatorias y una neurona en la salida. El factor de suavizado óptimo resultó en 3.0 para los vectores DT-DWT y 3.5 para los DT-WPT.

Una vez entrenada la red, ésta es almacenada para después ser utilizada en el proceso de clasificación para obtener a la salida del clasificador un vector (1x160) que contiene el resultado de la clasificación de cada una de las muestras y de igual manera se obtiene la matriz de clasificación, la tasa de acierto y la tasa de error.

3.4 CLASIFICADOR CON MÁQUINAS DE VECTORES SOPORTE (SVM)

El sistema de clasificación a utilizar en esta sección son las “máquinas de vectores soporte” ó SVM, en su versión más simple las SVM lineal y otras dos un poco más complejas que hacen uso de funciones kernels polinomial y funciones kernel de base radial ó RBF.

En general, para los tres tipos de clasificadores SVM, las muestras para el entrenamiento $x_1 \dots x_{160}$ se organizan en una matriz bidimensional 12×160 , en donde 12 corresponde al número de características y 160 es el número de muestras (ver figura 16) y se construye un vector 1×160 para etiquetar la clase de movimiento (1, 2, 3 ó 4) a la cual pertenece cada muestra de entrenamiento.

Tanto para el caso lineal como para el caso no lineal se parte de un conjunto de entrenamiento:

$$s = \{(x_i, y_i) : i = 1, \dots, 160\}$$

Teniendo ya las matrices para entrenamiento, se puede construir el clasificador Lineal, Polinomial o RBF.

Ahora, hay que definir el número de clasificadores que se van a construir ya que este es un problema multiclase (ver apartado 2.3.5). Según las ecuaciones descritas en el método “uno contra uno”, la cantidad de clasificadores parciales son:

$$m = \frac{\ell(\ell-1)}{2} = \frac{4(4-1)}{2} = 6$$

Por lo tanto la SVM tiene que realizar el aprendizaje para 6 clasificadores parciales para seis posibles combinaciones entre las cuatro clases, como se muestra a continuación:

Clasificador 1: Clase 1 Vs Clase 2

Clasificador 2: Clase 1 Vs Clase 3

Clasificador 3: Clase 1 Vs Clase 4

Clasificador 4: Clase 2 Vs Clase 3

Clasificador 5: Clase 2 Vs Clase 4

Clasificador 6: Clase 3 Vs Clase 4

➤ **Matriz de descomposición.**

Siendo $(t_j = +1)$ las etiquetas positivas donde $j \in \{1, \dots, 4\}$, y $(t_p = -1)$ las etiquetas negativas donde $p \in \{1, \dots, 4\}$ y $p \neq j$ y $(t_r = 0)$.

La matriz de descomposición para 4 clases que va a ser utilizada en el caso de separación lineal y el no lineal, estaría dada por:

$$D_{1-v-1} = \begin{cases} +1 & -1 & 0 & 0 \\ +1 & 0 & -1 & 0 \\ +1 & 0 & 0 & -1 \\ 0 & +1 & -1 & 0 \\ 0 & +1 & 0 & -1 \\ 0 & 0 & +1 & -1 \end{cases}$$

3.4.1 Asumiendo datos linealmente separables

Para realizar un entrenamiento lineal, suponemos que los datos de entrenamiento pueden ser separados linealmente (Capítulo 2, figura 8), (caso que no lo es para los datos utilizados en este trabajo)¹¹, esto se hace para que el resultado del método lineal presente un buen entrenamiento basado en información veraz.

Ahora, con cada una de las 6 máquinas biclasificadoras ya entrenadas, las cuales emiten una respuesta numérica al desarrollar las ecuaciones (2.38) y (2.39) vistas en el capítulo 2:

$$g(\vec{x}) = \sum_{i=1}^{VS} [\alpha_i y_i \langle \vec{x}_i \cdot \vec{x} \rangle] + b$$

¹¹ Se realiza clasificación lineal con SVM con objeto de mostrar la diferencia en los resultados cuando se utiliza un método lineal para clasificar datos no linealmente separables.

$$f(\vec{x}) = \text{sign}\left(\sum_{i=1}^{VS} [\alpha_i y_i \langle \vec{x}_i \cdot \vec{x} \rangle] + b\right)$$

Se obtiene $Z_i = g_i(\vec{x})$ y $S^i = f_i(\vec{x}) = \text{sign}(g_i(\vec{x}))$ que son las predicciones y los signos de cada entrada respectivamente. Teniendo esta información se procede al método de reconstrucción por votación según:

$$\theta(S^i) = \begin{cases} y_i & \text{si } S^i = +1 \\ 0 & \text{si } S^i = -1 \end{cases}$$

Entonces, al seguir el esquema de votación por mayoría simple puede ser clasificado un nuevo dato que se introduzca en la SVM.

3.4.2 Asumiendo datos linealmente no separables

En este trabajo y para la SVM en particular, se implementaron las funciones kernel polinomial y kernel RBF; consiguiendo con esto un mapeo de los datos a una dimensión mucho mayor que la original y así poder realizar una separación lineal.

Ahora, con cada una de las 6 máquinas biclasificadoras ya entrenadas, las cuales emiten una respuesta numérica al desarrollar las ecuaciones (2.52) y (2.53) vistas en el capítulo 2:

$$g(\vec{x}) = \sum_{i=1}^{VS} [\alpha_i y_i k(\vec{x}_i \cdot \vec{x})] + b$$

$$f(\vec{x}) = \text{sign}\left(\sum_{i=1}^{VS} [\alpha_i y_i k(\vec{x}_i \cdot \vec{x})] + b\right)$$

➤ **Clasificación con kernel polinomial**

En el caso de utilizar la función kernel Polinomial,

$$k(\vec{x}, \vec{y}) = (\vec{x} \cdot \vec{y} + c)^d \quad \text{para } c > 0$$

es posible variar el parámetro *Degree* (d) o Grados para tener un mejor resultado, ya que la curvatura del margen separador depende de este valor. Por lo tanto se realizaron varios experimentos variando este parámetro en un amplio rango, desde 0,1 a 4 donde se concluyó que los mejores resultados estaban entre los rangos 0,5 y 3,0 para los datos de entrenamiento utilizados en este trabajo.

➤ **Clasificación con kernel RBF**

Al utilizar la función kernel RBF,

$$k(\vec{x}, \vec{y}) = e^{-\frac{|\vec{x}-\vec{y}|^2}{2\sigma^2}}$$

se varía el parámetro Gamma (σ) con el cual puede variarse el área de error posible en la selección entre las clases, que después de varias pruebas se registraron los resultados entre los rangos 0.001 y 0.05.

Teniendo entrenada la SVM con el kernel seleccionado, ya es posible realizar ejercicios de clasificación para la validación del mismo, ingresando ya sea un solo vector o una matriz que contenga el conjunto de vectores característica.

En resumen, las SVM se definen mediante la selección de la constante de penalización C , el valor del parámetro σ para el caso del kernel RBF y el valor del parámetro d para el caso del kernel Polinomial. Al variar estos parámetros lo que se busca es tener un mejor desempeño de la SVM.

3.5 CLASIFICADOR CON UNA RED FUZZY ARTMAP.

En la red Fuzzy ARTMAP, a diferencia de los clasificadores anteriores, los datos de entrenamiento son definidos como una matriz $b \times a$, y la definición de la clase a la que pertenecen se hace con un vector $b \times 1$. En donde $b=160$ y hace referencia al número de muestras y $a=12$ características.

Primero los datos son normalizados. La normalización se hace por columna, es decir, se busca el máximo valor de cada columna y todos los valores de la misma se dividen por dicho valor. De esta forma se consigue que cada columna contenga valores entre "0" y "1". Después al vector de entradas se le une el de su complemento a uno, para formar un vector de tamaño doble $b \times 2a$ (160 x 24). Esto tiene unas ventajas prácticas, entre las que se encuentran la no proliferación de categorías. Además, permitirá usar el operador lógico (\wedge) para las operaciones que efectuarán los distintos módulos. La entrada complementada se presenta a la red Fuzzy ART A y este algoritmo de agrupamiento es el que se encarga en primer lugar de elegir cuál es la categoría más cercana a la entrada presentada.

Después se evalúa si dicha categoría tiene un valor de ajuste lo suficientemente grande como para superar un umbral elegido o "factor de vigilancia" que mediante ensayo y error fue definido como 0.4. Si es superado este valor se considera que la entrada pertenece a esa categoría. En caso de que no lo supere, elige la siguiente categoría más cercana, y repite la evaluación. Si ninguna de las características existentes cumple con este "criterio de ajuste", se deja dicha muestra sin clasificar, en cuyo caso el índice es cero (0). En resumen, si se encuentra alguna que cumple con el criterio, se presenta su salida al modulo de enlace y éste se encargará de decidir si la salida propuesta coincide con la deseada. En caso de que sea así, se produciría el aprendizaje de la nueva categoría, en caso contrario, se lanza la búsqueda de una nueva categoría.

4 PRUEBAS Y EVALUACIÓN DE LOS CLASIFICADORES

Con cada uno de los clasificadores se realizaron diversas pruebas hasta conseguir el mínimo valor de error posible utilizando los datos de validación. Estas pruebas fueron realizadas variando los parámetros de cada clasificador antes del entrenamiento, validando el desempeño con nuevos datos y posteriormente recopilando los resultados en tablas que facilitan su análisis y la comparación de todos los métodos de clasificación.

Primero se expondrán en el apartado 4.1 los resultados de la clasificación obtenidos con cada uno de los métodos, utilizando los dos tipos de vectores híbridos de manera independiente. Segundo, para cada método será seleccionado el o los parámetros con los que el clasificador muestre mejor desempeño. Tercero, se llevará a cabo la respectiva evaluación en el apartado 4.2, teniendo en cuenta los aspectos definidos en el apartado 1.3.2 que son: precisión, velocidad de clasificación, potencia del clasificador y flexibilidad.

4.1 PRUEBAS

Los resultados de las pruebas realizadas para cada uno de los clasificadores son plasmados en los gráficos y tablas de este apartado. Los gráficos muestran en su eje "x" las variaciones de los parámetros (correspondientes a cada clasificador) y en el eje "y" el porcentaje de error obtenido para cada paciente. Las tablas muestran con más detalle estos resultados, adicionando el error promedio por cada una de las variaciones de los parámetros.

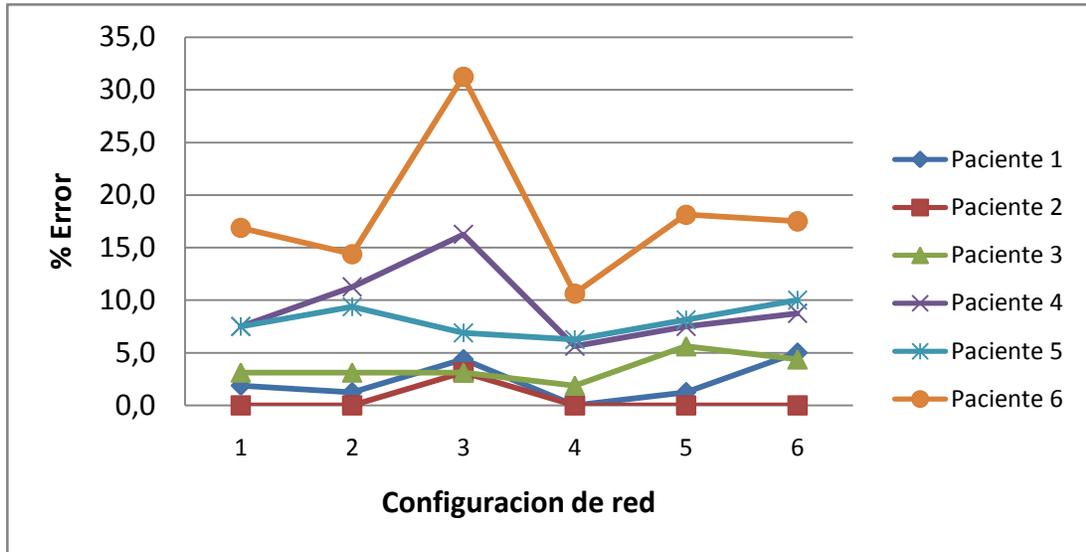
4.1.1 Perceptrón multicapa (MLP)

Se probaron seis configuraciones de red, en las que se variaron el número de capas ocultas y la cantidad de neuronas en ellas. Para un mejor entendimiento de las graficas y las tablas 1 y 2, se numeraron de 1 a 6 las diferentes configuraciones de red que se describen a continuación:

1. Una capa oculta con seis neuronas, con función de activación tangencial.
2. Una capa oculta con ocho neuronas, con función de activación tangencial.
3. Una capa oculta con doce neuronas, con función de activación tangencial.
4. Dos capas ocultas con seis neuronas en cada una de ellas, con funciones de activación tangenciales.
5. Dos capas ocultas con ocho neuronas en cada una de ellas, con funciones de activación logarítmicas.
6. Dos capas ocultas con doce y seis neuronas respectivamente, con funciones de activación logarítmicas.

En las gráficas y tablas 1 y 2 se muestran los resultados de la clasificación de los datos con MLP.

➤ **Clasificación utilizando la técnica DT-DWT**



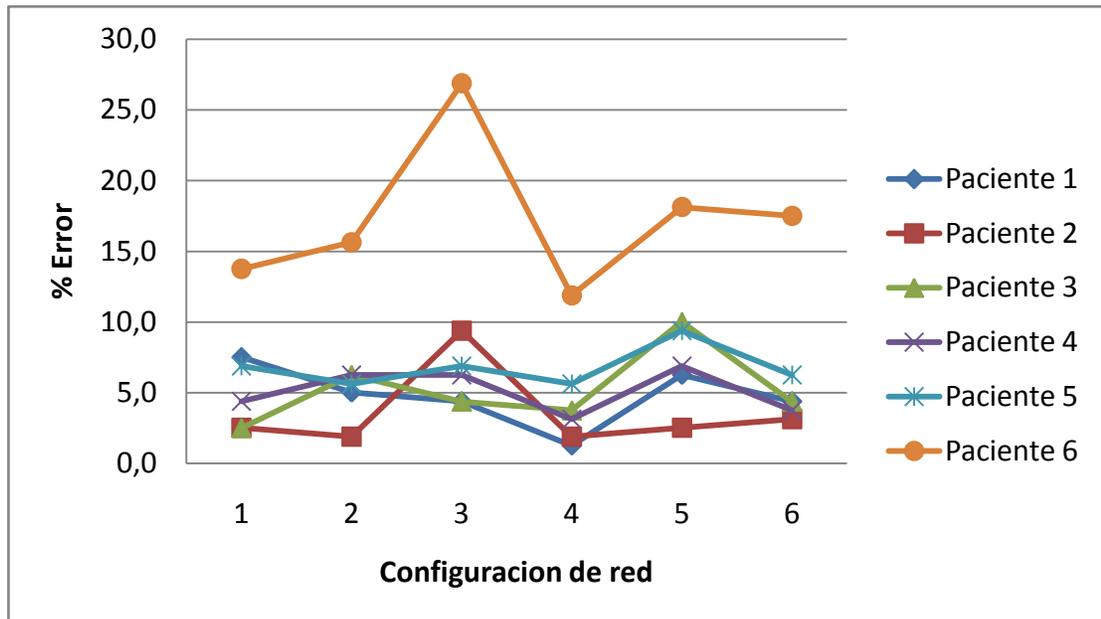
Grafica 1. Resultados de la clasificación. Técnica: DT – DWT, Método: MLP

	1	2	3	4	5	6
Paciente 1	1,875	1,250	4,375	0,000	1,250	5,000
Paciente 2	0,000	0,000	3,125	0,000	0,000	0,000
Paciente 3	3,125	3,125	3,125	1,875	5,625	4,375
Paciente 4	7,500	11,250	16,250	5,625	7,500	8,750
Paciente 5	7,500	9,375	6,875	6,250	8,125	10,000
Paciente 6	16,875	14,375	31,250	10,625	18,125	17,500
Promedio	6,146	6,563	10,833	4,063	6,771	7,604

Tabla 1. Porcentajes de error de clasificación. Técnica: DT – DWT, Método: MLP.

De las configuraciones de red implementadas utilizando la técnica DT – DWT, la mejor es la número 4, que además de presentar el menor promedio de error es en la que mayor número de pacientes tienen el menor porcentaje.

➤ **Clasificación utilizando la técnica DT-WPT**



Grafica 2. Resultados de la clasificación. Técnica: DT – WPT, Método: MLP

	1	2	3	4	5	6
Paciente 1	7,500	5,000	4,375	1,250	6,250	4,375
Paciente 2	2,500	1,875	9,375	1,875	2,500	3,125
Paciente 3	2,500	6,250	4,375	3,750	10,000	4,375
Paciente 4	4,375	6,250	6,250	3,125	6,875	3,750
Paciente 5	6,875	5,625	6,875	5,625	9,375	6,250
Paciente 6	13,750	15,625	26,875	11,875	18,125	17,500
Promedio	6,250	6,771	9,688	4,583	8,854	6,563

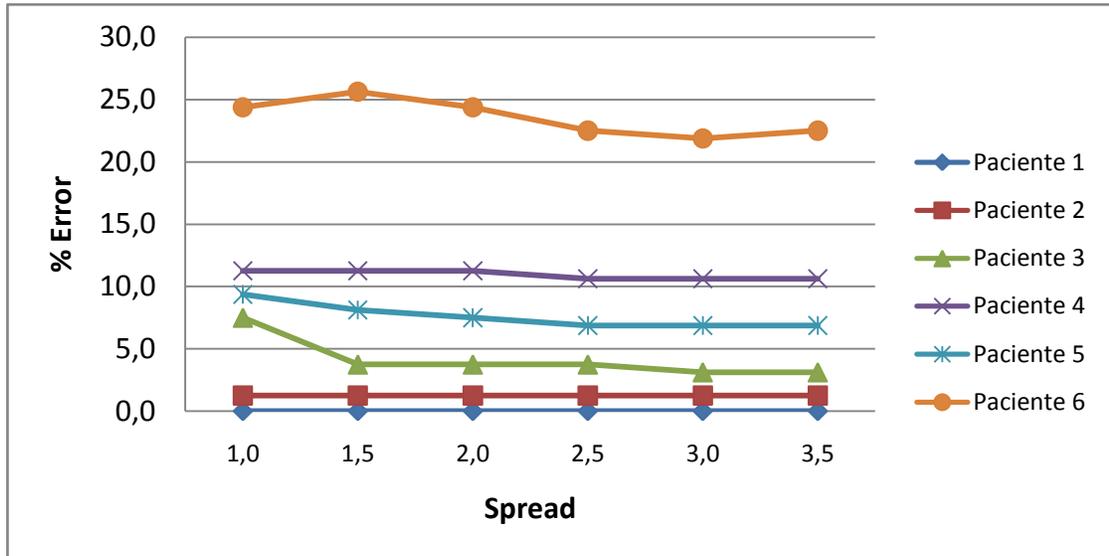
Tabla 2. Porcentajes de error de clasificación. Técnica: DT – WPT, Método: MLP

Con la técnica DT – WPT, la mejor configuración también es la número 4.

4.1.2 Red neuronal probabilística (PNN)

En la PNN, el único valor o parámetro que es posible manipular es “*spread*”. Se realizaron variaciones de este parámetro en un rango de 1 a 3.5 ya que fuera de este intervalo los resultados arrojan valores de error muy elevados. Los resultados se muestran en las gráficas y tablas 3 y 4.

➤ **Clasificación utilizando la técnica DT-DWT**



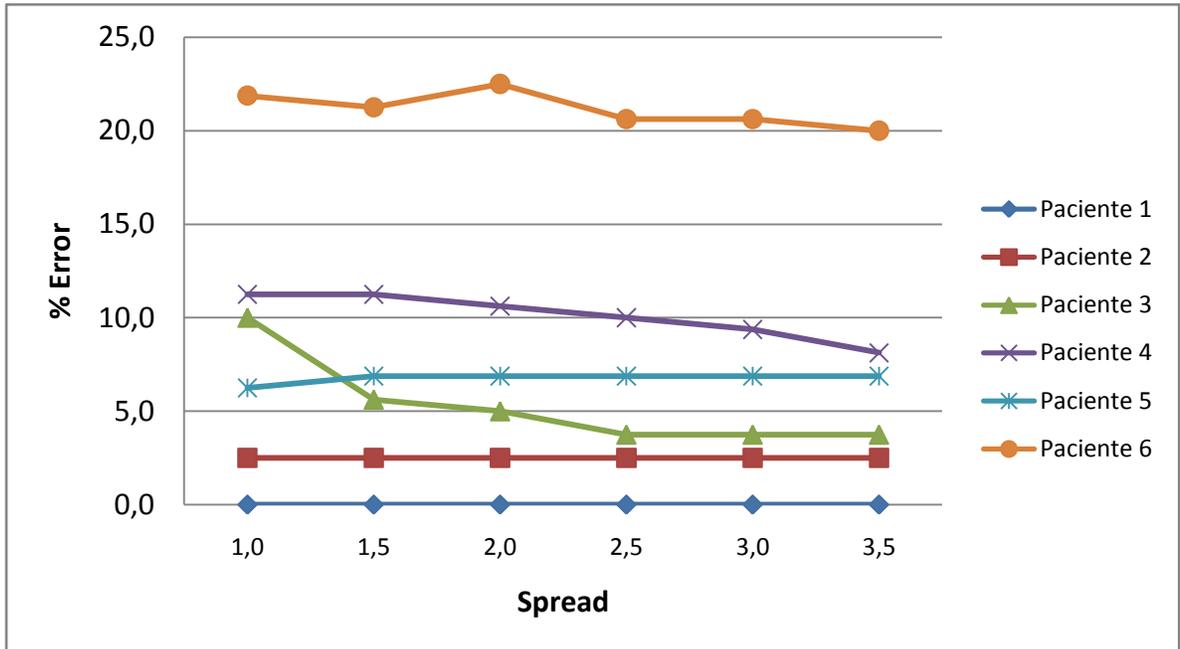
Grafica 3. Resultados de la clasificación. Técnica: DT – DWT, Método: PNN

	Spread =1,0	Spread =1,5	Spread =2,0	Spread =2,5	Spread =3,0	Spread =3,5
Paciente 1	0,000	0,000	0,000	0,000	0,000	0,000
Paciente 2	1,250	1,250	1,250	1,250	1,250	1,250
Paciente 3	7,500	3,750	3,750	3,750	3,125	3,125
Paciente 4	11,250	11,250	11,250	10,625	10,625	10,625
Paciente 5	9,375	8,125	7,500	6,875	6,875	6,875
Paciente 6	24,375	25,625	24,375	22,500	21,875	22,500
Promedio	8,958	8,333	8,021	7,500	7,292	7,396

Tabla 3. Porcentajes de error de clasificación. Técnica: DT – DWT, Método: PNN

Utilizando la técnica DT-DWT, el mejor clasificador es el obtenido después de entrenar la red con *spread* igual a 3.

➤ **Clasificación utilizando la técnica DT-WPT**



Grafica 4. Resultados de la clasificación. Técnica: DT – WPT, Método: PNN

	Spread =1,0	Spread =1,5	Spread =2,0	Spread =2,5	Spread =3,0	Spread =3,5
Paciente 1	0,000	0,000	0,000	0,000	0,000	0,000
Paciente 2	2,500	2,500	2,500	2,500	2,500	2,500
Paciente 3	10,000	5,625	5,000	3,750	3,750	3,750
Paciente 4	11,250	11,250	10,625	10,000	9,375	8,125
Paciente 5	6,250	6,875	6,875	6,875	6,875	6,875
Paciente 6	21,875	21,250	22,500	20,625	20,625	20,000
Promedio	8,646	7,917	7,917	7,292	7,188	6,875

Tabla 4. Porcentajes de error de clasificación. Técnica: DT – WPT, Método: PNN

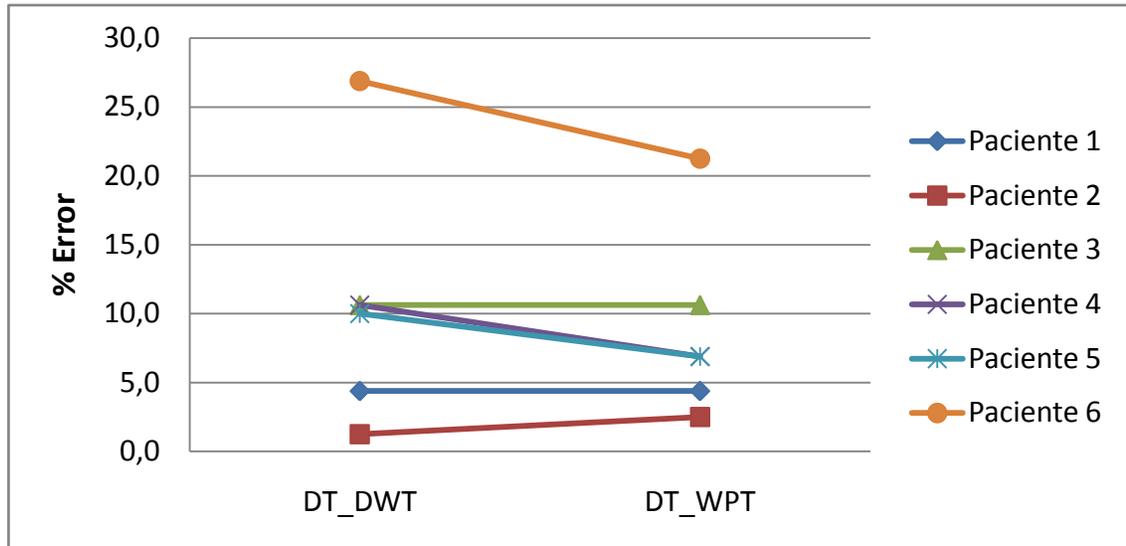
En este caso se obtienen mejores resultados con *spread* igual a 3.5.

4.1.3 SVM asumiendo separación de datos lineal.

SVM lineal no tiene ningún parámetro que modificar, es decir, es un método de solución única. Esto se debe a que él mismo se encarga de hallar los mejores

parámetros para su configuración. En la gráfica y la tabla 5 se muestran los resultados de la clasificación utilizando las dos técnicas DT-DWT y DT - WPT.

➤ **Clasificación utilizando la técnica DT-DWT y DT-WPT**



Grafica 5. Resultados de la clasificación. Técnica: DT - DWT y DT - WPT, Método: SVM LINEAL

	DT_DWT	DT_WPT
Paciente 1	4,375	4,375
Paciente 2	1,250	2,500
Paciente 3	10,625	10,625
Paciente 4	10,625	6,875
Paciente 5	10,000	6,875
Paciente 6	26,875	21,250
Promedio	10,625	8,750

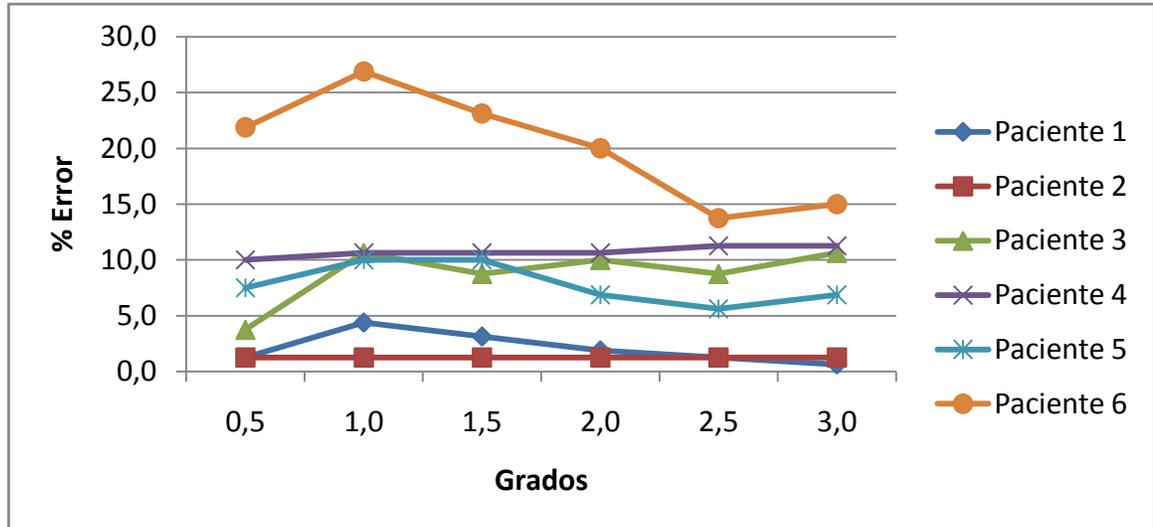
Tabla 5. Porcentajes de error de clasificación. Técnica: DT - WPT y DT - WPT, Método: SVM LINEAL

4.1.4 SVM con función kernel polinomial

En la SVM con función kernel polinomial, a la que se hará referencia como SVM Polinomial, se realizaron variaciones del parámetro “Grados” en un rango de 0,5 a

3 ya que fuera de este intervalo los resultados no presentan mejoras notables y el tiempo de entrenamiento es muy largo. Los resultados se muestran en las gráficas y tablas 6 y 7.

➤ **Clasificación utilizando la técnica DT-DWT**



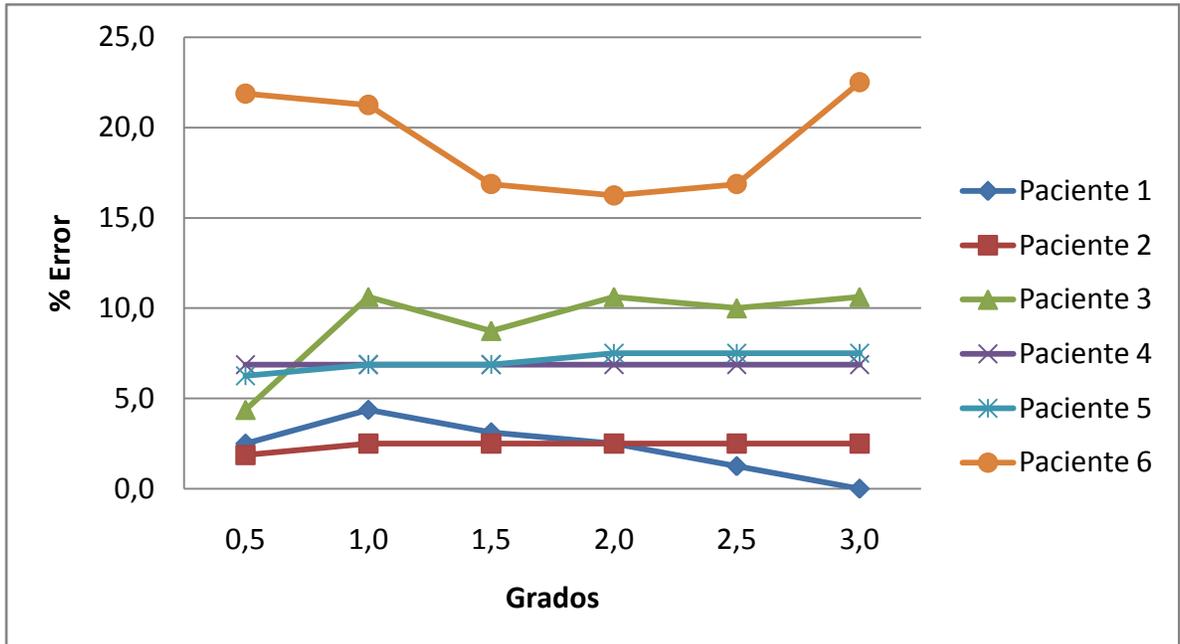
Grafica 6. Resultados de la clasificación. Técnica: DT – DWT, Método: SVM POLINOMIAL

	Grado =0,5	Grado =1,0	Grado =1,5	Grado =2,0	Grado =2,5	Grado = 3,0
Paciente 1	1,250	4,375	3,125	1,875	1,250	0,625
Paciente 2	1,250	1,250	1,250	1,250	1,250	1,250
Paciente 3	3,750	10,625	8,750	10,000	8,750	10,625
Paciente 4	10,000	10,625	10,625	10,625	11,250	11,250
Paciente 5	7,500	10,000	10,000	6,875	5,625	6,875
Paciente 6	21,875	26,875	23,125	20,000	13,750	15,000
Promedio	7,604	10,625	9,479	8,438	6,979	7,604

Tabla 6. Porcentajes de error de clasificación. Técnica: DT – DWT, Método: SVM POLINOMIAL

A pesar que el menor promedio de error corresponde a Grados igual a 2.5, es preferible utilizar Grados igual a 0.5, debido a que la clasificación de las señales de todos los pacientes exceptuando el número 6, es mejor con este valor.

➤ **Clasificación utilizando la técnica DT-WPT**



Grafica 7. Resultados de la clasificación. Técnica: DT – WPT, Método: SVM POLINOMIAL

	Grados =0,5	Grados =1,0	Grados =1,5	Grados =2,0	Grados =2,5	Grados =3,0
Paciente 1	2,500	4,375	3,125	2,500	1,250	0,000
Paciente 2	1,875	2,500	2,500	2,500	2,500	2,500
Paciente 3	4,375	10,625	8,750	10,625	10,000	10,625
Paciente 4	6,875	6,875	6,875	6,875	6,875	6,875
Paciente 5	6,250	6,875	6,875	7,500	7,500	7,500
Paciente 6	21,875	21,250	16,875	16,250	16,875	22,500
Promedio	7,292	8,750	7,500	7,708	7,500	8,333

Tabla 7. Porcentajes de error de clasificación. Técnica: DT – WPT, Método: SVM POLINOMIAL

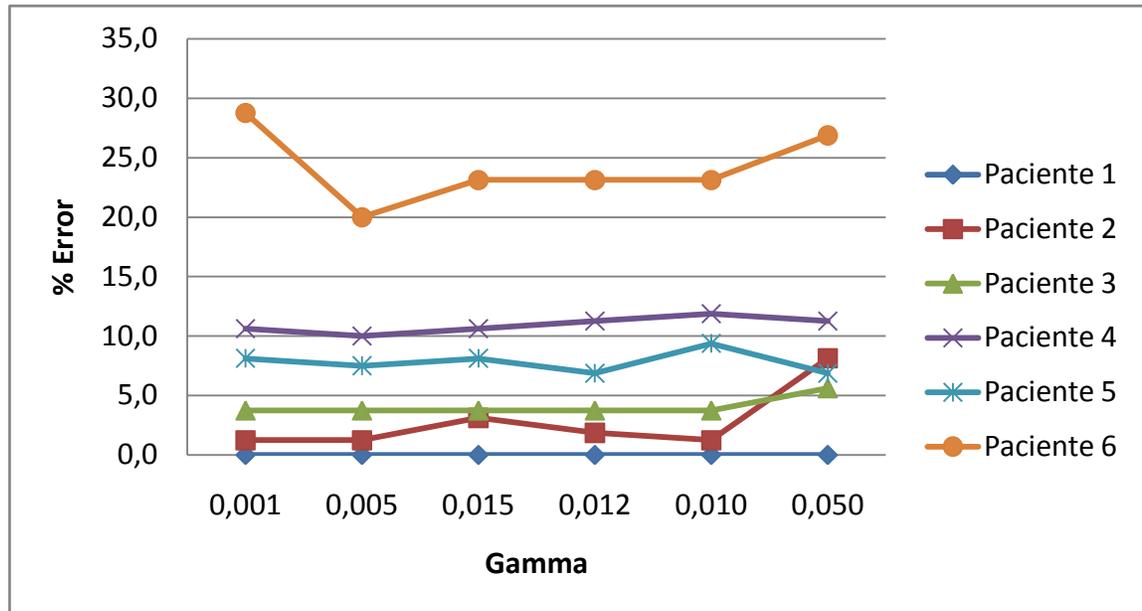
Después de validar el clasificador con la técnica DT – WPT el mejor resultado se obtiene con Grados igual a 0.5.

4.1.5 SVM con kernel función de base radial (RBF)

Con la SVM con funciones de base radial, a la que se hará referencia como SVM RBF, es posible variar los resultados con el cambio del parámetro “Gamma” σ .

Este parámetro se fijó en 6 valores (dentro del rango 0,001 – 0,050; para otros valores los resultados tienen un porcentaje de error muy alto) tanto para las pruebas con técnicas DT-DWT como para DT-WPT, como se muestra en la gráfica y tabla 8 y 9 respectivamente.

➤ **Clasificación utilizando la técnica DT-DWT**



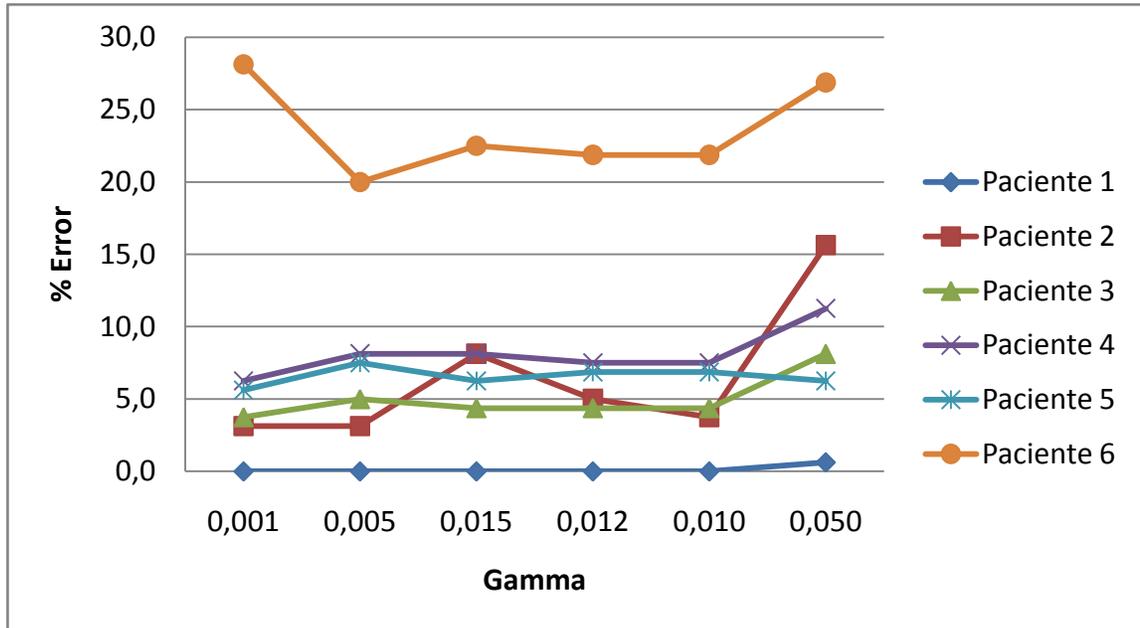
Grafica 8. Resultados de la clasificación. Técnica: DT – DWT, Método: SVM RBF

	Gamma= 0,001	Gamma= 0,005	Gamma= 0,015	Gamma= 0,012	Gamma= 0,010	Gamma= 0,050
Paciente 1	0,000	0,000	0,000	0,000	0,000	0,000
Paciente 2	1,250	1,250	3,125	1,875	1,250	8,125
Paciente 3	3,750	3,750	3,750	3,750	3,750	5,625
Paciente 4	10,625	10,000	10,625	11,250	11,875	11,250
Paciente 5	8,125	7,500	8,125	6,875	9,375	6,875
Paciente 6	28,750	20,000	23,125	23,125	23,125	26,875
Promedio	8,750	7,083	8,125	7,813	8,229	9,792

Tabla 8. Porcentajes de error de clasificación. Técnica: DT – DWT, Método: SVM RBF

Utilizando la técnica DT – DWT con el método SVM RBF el mejor resultado obtenido es con Gamma igual a 0,005.

➤ **Clasificación utilizando la técnica DT-WPT**



Grafica 9. Resultados de la clasificación. Técnica: DT – WPT, Método: SVM RBF

	Gamma= 0,001	Gamma= 0,005	Gamma= 0,015	Gamma= 0,012	Gamma= 0,010	Gamma= 0,050
Paciente 1	0,000	0,000	0,000	0,000	0,000	0,625
Paciente 2	3,125	3,125	8,125	5,000	3,750	15,625
Paciente 3	3,750	5,000	4,375	4,375	4,375	8,125
Paciente 4	6,250	8,125	8,125	7,500	7,500	11,250
Paciente 5	5,625	7,500	6,250	6,875	6,875	6,250
Paciente 6	28,125	20,000	22,500	21,875	21,875	26,875
Promedio	7,813	7,292	8,229	7,604	7,396	11,458

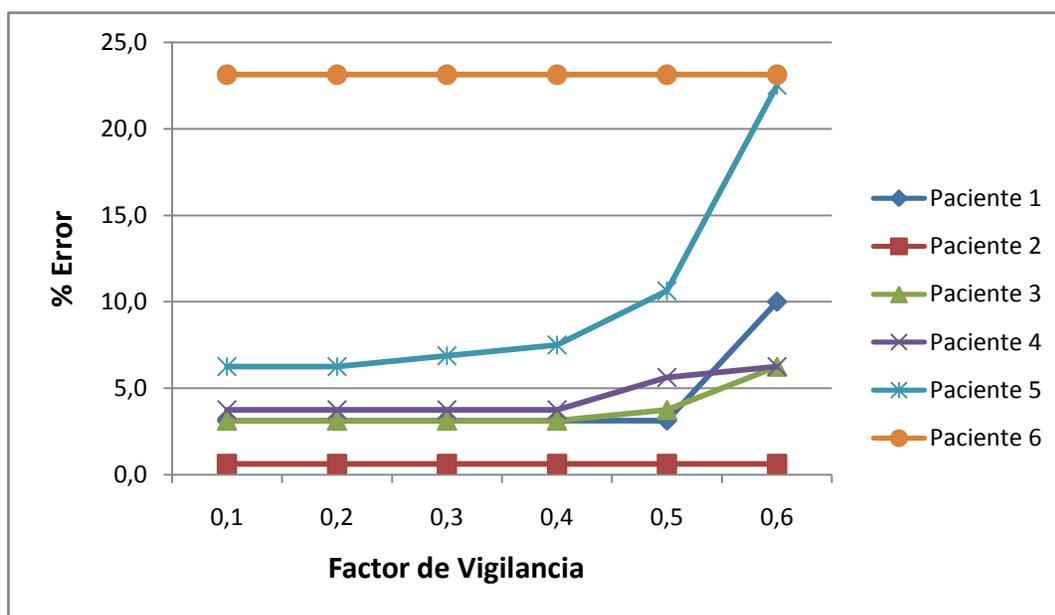
Tabla 9. Porcentajes de error de clasificación. Técnica: DT – WPT, Método: SVM RBF

Con la técnica DT – WPT con el método SVM RBF el mejor resultado obtenido es con Gamma igual a 0,001

4.1.6 Fuzzy ARTMAP

Los resultados obtenidos al utilizar Fuzzy ARTMAP y al variar el factor de vigilancia (FV) dentro del rango 0,1 y 0,6 para las técnicas DT-DWT y DT-WPT, se muestran en la gráficas y las tablas 10 y 11 respectivamente.

➤ **Clasificación utilizando la técnica DT-DWT**



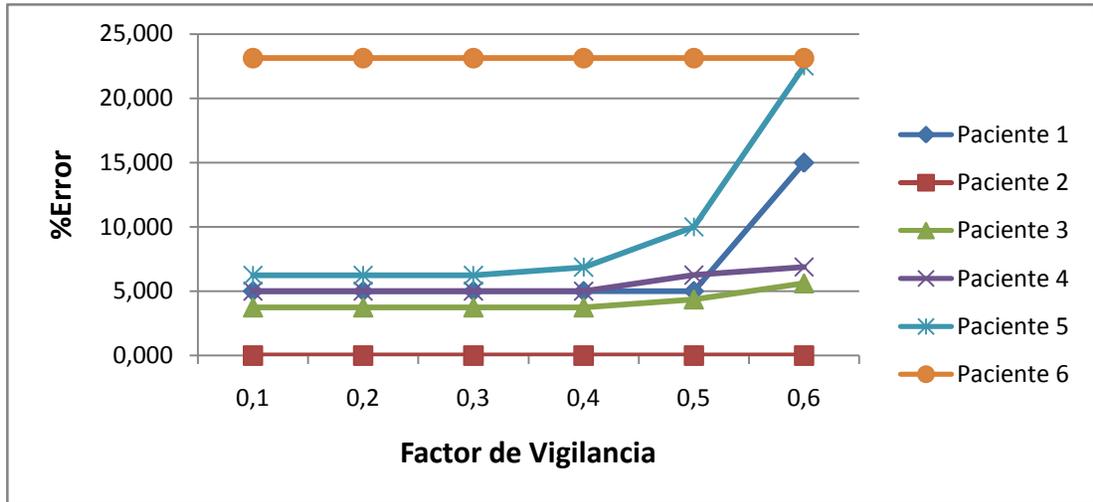
Grafica 10. Resultados de la clasificación. Técnica: DT – DWT, Método: Fuzzy ARTMAP

	FV = 0,1	FV = 0,2	FV = 0,3	FV = 0,4	FV = 0,5	FV = 0,6
Paciente 1	3,125	3,125	3,125	3,125	3,125	10,000
Paciente 2	0,625	0,625	0,625	0,625	0,625	0,625
Paciente 3	3,125	3,125	3,125	3,125	3,750	6,250
Paciente 4	3,750	3,750	3,750	3,750	5,625	6,250
Paciente 5	6,250	6,250	6,875	7,500	10,625	22,500
Paciente 6	23,125	23,125	23,125	23,125	23,125	23,125
Promedio	6,667	6,667	6,771	6,875	7,813	11,458

Tabla 10. Porcentajes de error de clasificación. Técnica: DT – DWT, Método: Fuzzy ARTMAP

Para la clasificación utilizando la técnica DT-DWT el mejor resultado se obtuvo con FV igual a 0,1 y 0,2 ya que en los dos casos se presentaron iguales resultados.

➤ **Clasificación utilizando la técnica DT-WPT**



Grafica 11. Resultados de la clasificación. Técnica: DT – WPT, Método: Fuzzy ARTMAP

	FV = 0,1	FV = 0,2	FV = 0,3	FV = 0,4	FV = 0,5	FV = 0,6
Paciente 1	5,000	5,000	5,000	5,000	5,000	15,000
Paciente 2	0,000	0,000	0,000	0,000	0,000	0,000
Paciente 3	3,750	3,750	3,750	3,750	4,375	5,625
Paciente 4	5,000	5,000	5,000	5,000	6,250	6,875
Paciente 5	6,250	6,250	6,250	6,875	10,000	22,500
Paciente 6	23,125	23,125	23,125	23,125	23,125	23,125
Promedio	7,188	7,188	7,188	7,292	8,125	12,188

Tabla 11. Porcentajes de error de clasificación. Técnica: DT – WPT, Método: Fuzzy ARTMAP

Con la utilización de la técnica DT-WPT el mejor resultado se obtuvo con un factor de vigilancia de 0,1 y 0,2.

4.2 EVALUACION

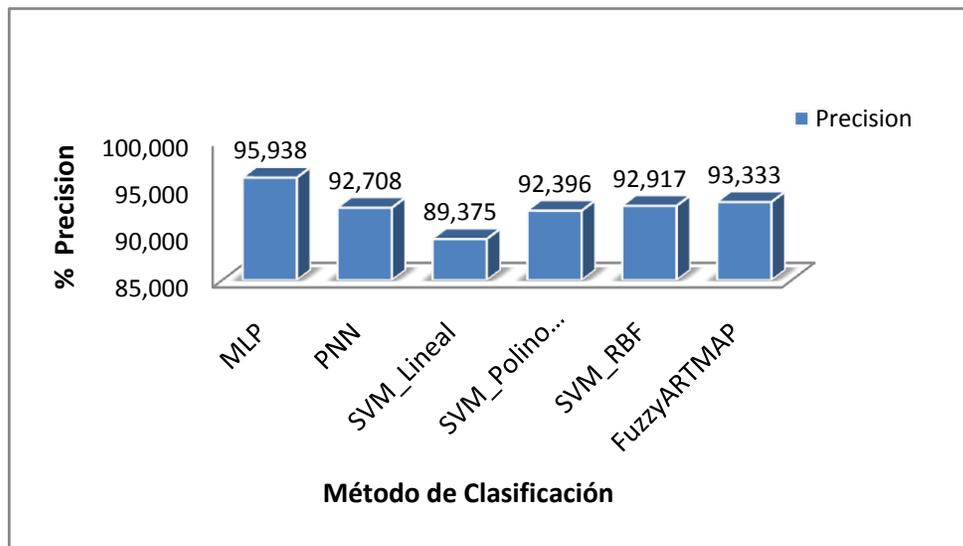
Para evaluar y así determinar la bondad de los clasificadores implementados se tendrán en cuenta la precisión o exactitud, la velocidad o tiempo de clasificación, la potencia y la flexibilidad con respecto al desempeño presentado utilizando los datos designados para la validación. Se tendrán solamente en cuenta los mejores

resultados de cada uno de los métodos de clasificación utilizados en este trabajo y que fueron expuestos en la sección anterior.

4.2.1 Exactitud o precisión

El clasificador es el encargado de identificar el movimiento que el paciente desea ejecutar, es decir, debe generar la señal correcta para controlar la prótesis, por lo tanto debe tener un alto grado de precisión o exactitud. Para evaluar este parámetro, se realizará una comparación entre los diferentes métodos empleados, teniendo en cuenta el acierto promedio por método llamado en este caso “precisión” (la precisión se evalúa de 0% a 100%, y para hallarla se resta el porcentaje de error promedio obtenido al 100%) y se muestran en las gráficas 12 y 13. Para mayor detalle se muestran también las tablas 12 y 13, resaltando en color rojo los valores que representen el menor porcentaje de error por paciente.

➤ **Precisión de la clasificación utilizando la técnica DT-DWT**



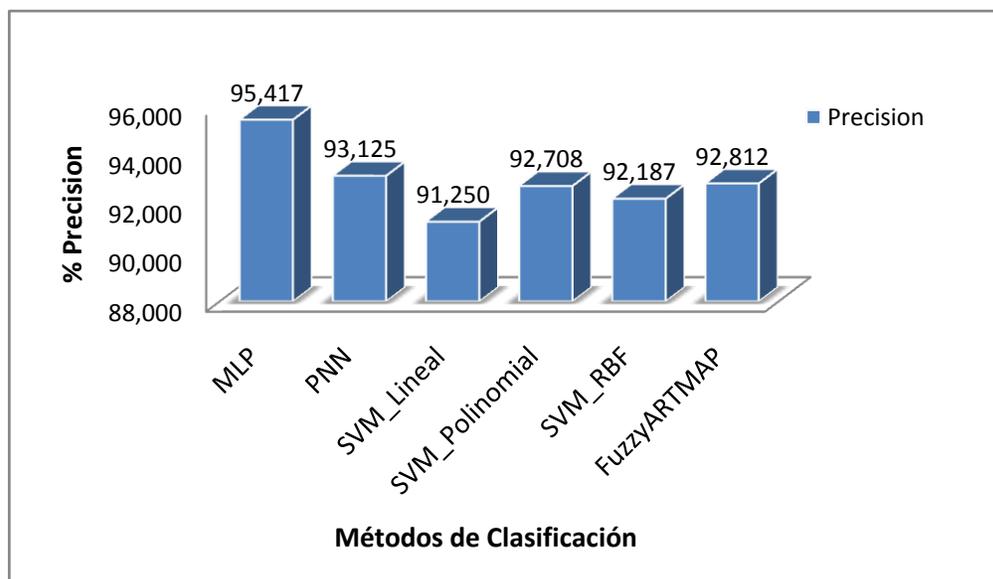
Grafica 12. Comparación de la precisión de los clasificadores con la técnica DT_DWT

	MLP	PNN	SVM_Lineal	SVM_Polinomial	SVM_RBF	FuzzyARTMAP
Paciente 1	0,000	0	4,375	1,25	0	3,125
Paciente 2	0,000	1,25	1,25	1,25	1,25	0,625

Paciente 3	1,875	3,125	10,625	3,75	3,75	3,125
Paciente 4	5,625	10,625	10,625	10	10	3,75
Paciente 5	6,250	6,875	10	7,5	7,5	6,25
Paciente 6	10,650	21,875	26,875	21,875	20	23,125
Error Promedio	4,063	7,292	10,625	7,604	7,083	6,667
Precisión	95,938	92,708	89,375	92,396	92,917	93,333

Tabla 12. Comparación de la precisión de los clasificadores con la técnica DT_DWT (valores en %).

➤ **Precisión de la clasificación utilizando la técnica DT-WPT**



Grafica 13. Comparación de la precisión de los clasificadores con la técnica DT_WPT

	MLP	PNN	SVM_Lineal	SVM_Polinomial	SVM_RBF	FuzzyARTMAP
Paciente 1	1,25	0	4,375	2,500	0	5,00
Paciente 2	1,875	2,5	2,5	1,875	3,125	0
Paciente 3	3,750	3,75	10,625	4,375	3,75	3,75
Paciente 4	3,125	8,125	6,875	6,875	6,25	5,00
Paciente 5	5,625	6,875	6,875	6,25	5,625	6,25
Paciente 6	11,875	20	21,25	21,875	28,125	23,125
Error Promedio	4,583	6,875	8,75	7,292	7,813	7,188
Precisión	95,417	93,125	91,250	92,708	92,187	92,812

Tabla 13. Comparación de la precisión de los clasificadores con la técnica DT_WPT (valores en %).

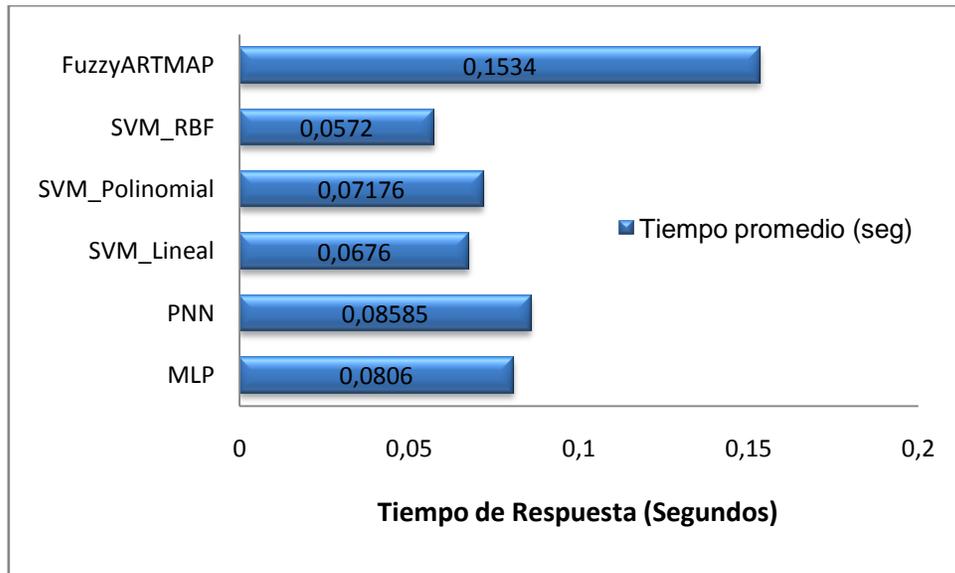
En la tabla 12 y 13 se puede ver que el clasificador MLP presenta un porcentaje de precisión más elevado que los otros métodos. Utilizando la técnica DT-DWT, cinco de los seis pacientes encuentran aquí la menor tasa de error, y con la técnica DT- WPT son cuatro los pacientes que obtienen mejores resultados, razón por la que se puede calificar como el clasificador más preciso entre los estudiados e implementados en este documento. En contraste el clasificador SVM – Lineal, es el más impreciso con las dos técnicas.

4.2.2 Tiempo de clasificación

Para ser usado en un sistema de control para prótesis de mano, el clasificador debe procesar y responder de la manera más rápida posible a las señales o estímulos que el ser humano o usuario le envíe. Los usuarios protésicos, generalmente no aceptan la demora mucho más larga que los 100ms entre el mandato (la contracción de músculo) y la activación de la prótesis en general. Los retardos en la reacción pueden generar molestias y si son muy extensos, la prótesis puede resultar no muy útil para el usuario.

El tiempo de procesamiento o clasificación, fue calculado en un PC con las siguientes características: Procesador Intel® Core™2 Duo, memoria de 2 GB, sistema operativo Windows Vista Ultimate.

➤ **Tiempo promedio de la clasificación utilizando la técnica DT-DWT**

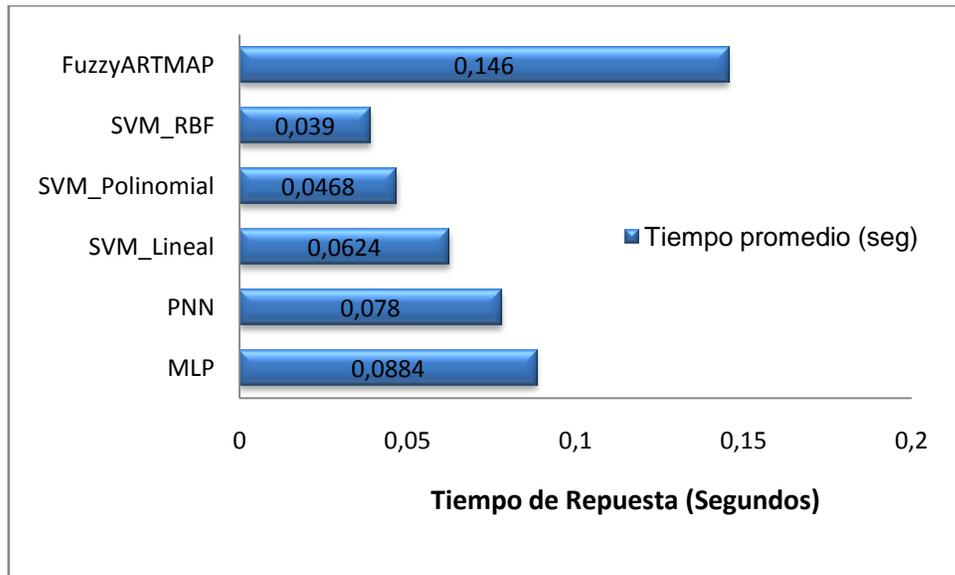


Grafica 14. Comparación de la velocidad promedio de los clasificadores con la técnica DT_DWT

	MLP	PNN	SVM_Lineal	SVM_Polinomial	SVM_RBF	FuzzyARTMAP
Paciente 1	0,0936	0,0936	0,0624	0,078	0,078	0,2028
Paciente 2	0,0936	0,0936	0,078	0,0624	0,0624	0,1872
Paciente 3	0,062	0,078	0,078	0,0624	0,0468	0,2028
Paciente 4	0,078	0,078	0,0312	0,078	0,0624	0,1092
Paciente 5	0,062	0,078	0,078	0,0624	0,0468	0,1092
Paciente 6	0,0936	0,0939	0,078	0,0936	0,0468	0,1092
Promedio	0,0806	0,08585	0,0676	0,07176	0,0572	0,1534

Tabla 14. Comparación de la velocidad promedio de los clasificadores con la técnica DT_DWT (valores en %).

➤ **Tiempo promedio de la clasificación utilizando la técnica DT-WPT**



Grafica 15. Comparación de la velocidad promedio de los clasificadores con la técnica DT_WPT

	MLP	PNN	SVM_Lineal	SVM_Polinomial	SVM_RBF	FuzzyARTMAP
Paciente 1	0,078	0,0624	0,0624	0,0312	0,0468	0,1248
Paciente 2	0,0936	0,078	0,0624	0,0312	0,0468	0,0936
Paciente 3	0,078	0,0936	0,0468	0,0624	0,0312	0,187
Paciente 4	0,0936	0,0624	0,0468	0,078	0,0468	0,1404
Paciente 5	0,0936	0,0936	0,0936	0,0468	0,0312	0,1404
Paciente 6	0,0936	0,078	0,0624	0,0312	0,0312	0,187
Promedio	0,0884	0,078	0,0624	0,0468	0,039	0,146

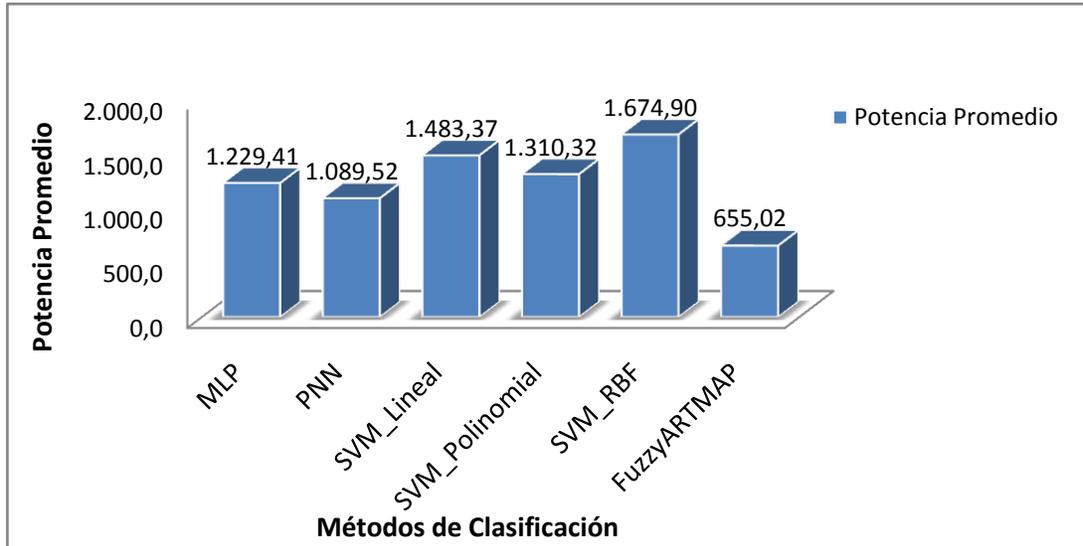
Tabla 15. Comparación de la velocidad promedio de los clasificadores con la técnica DT_WPT (valores en %).

La velocidad de los clasificadores en general es muy buena teniendo en cuenta que esta medida corresponde a la clasificación de 160 muestras. Sin embargo SVM- RBF presenta en los dos casos (gráficas y tablas 14 y 15) un procesamiento más rápido, mientras que Fuzzy Artmap es el método que tarda más en dar la respuesta.

4.2.3 Potencia

La Potencia de clasificación es calculada como la razón entre el porcentaje de precisión y el tiempo de clasificación o respuesta empleado.

➤ **Potencia de los clasificadores utilizando la técnica DT-DWT**

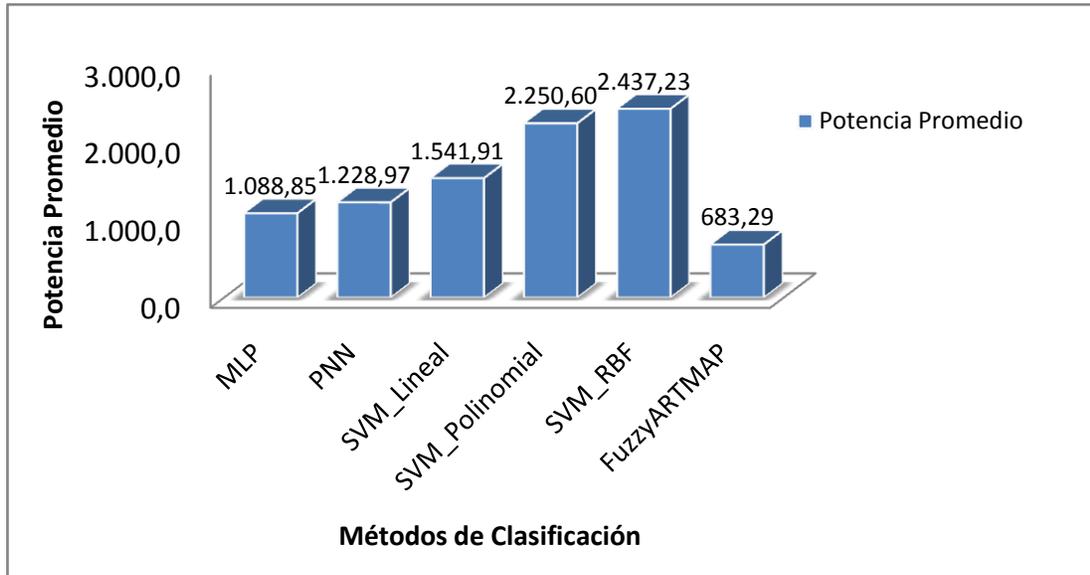


Grafica 16. Comparación de la potencia promedio de los clasificadores con la técnica DT_DWT

	MLP	PNN	SVM_Lineal	SVM_Polinomial	SVM_RBF	FuzzyARTMAP
Paciente 1	1.068,4	1.068,4	1.532,5	1.266,0	1.282,1	477,7
Paciente 2	1.068,4	1.055,0	1.266,0	1.582,5	1.582,5	530,8
Paciente 3	1.572,5	1.242,0	1.145,8	1.542,5	2.056,6	477,7
Paciente 4	1.209,9	1.145,8	2.864,6	1.153,8	1.442,3	881,4
Paciente 5	1.502,4	1.193,9	1.153,8	1.482,4	1.976,5	858,5
Paciente 6	954,9	832,0	937,5	834,7	1.709,4	704,0
Potencia Promedio	1.229,4	1.089,5	1.483,4	1.310,3	1.674,9	655,0

Tabla 16. Comparación de la potencia promedio de los clasificadores con la técnica DT_DWT (valores en %).

➤ **Potencia de los clasificadores utilizando la técnica DT-WPT**



Grafica 17. Comparación de la potencia promedio de los clasificadores con la técnica DT_WPT

	MLP	PNN	SVM_Lineal	SVM_Polinomial	SVM_RBF	FuzzyARTMAP
Paciente 1	1.266,0	1.602,6	1.532,5	3.125,0	2.136,8	761,2
Paciente 2	1.048,3	1.250,0	1.562,5	3.145,0	2.070,0	1.068,4
Paciente 3	1.234,0	1.028,3	1.909,7	1.532,5	3.084,9	514,7
Paciente 4	1.035,0	1.472,4	1.989,9	1.193,9	2.003,2	676,6
Paciente 5	1.008,3	994,9	994,9	2.003,2	3.024,8	667,7
Paciente 6	941,5	1.025,6	1.262,0	2.504,0	2.303,7	411,1
Potencia Promedio	1.088,9	1.229,0	1.541,9	2.250,6	2.437,2	683,3

Tabla 17. Comparación de la potencia promedio de los clasificadores con la técnica DT_WPT (valores en %).

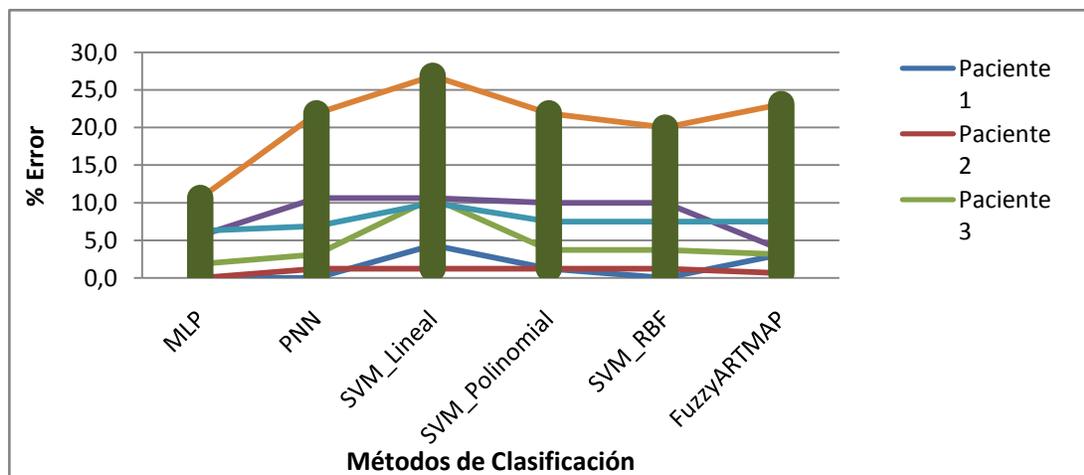
En las gráficas 16 y 17 se puede ver que los métodos SVM relacionan mejor la precisión y la velocidad de clasificación. Entre ellos el SVM – RBF tiene mayor potencia independientemente de la técnica utilizada.

4.2.4 Flexibilidad

Un clasificador debe ser flexible o adaptable, es decir, éste debe clasificar correctamente los datos de entrada, incluso si esos datos son considerablemente diferentes a los datos utilizados para el entrenamiento, además debe ser capaz de adaptarse a diferentes usuarios, sin que esto afecte su buen desempeño. Esta característica será medida comparando el desempeño del clasificador para los diferentes pacientes y teniendo en cuenta el rango en el que se encuentran los porcentajes de error cometidos.

En las tablas 18 y 19, se hace referencia a un rango de error, el cual corresponde al intervalo donde se encuentran los porcentajes de error de clasificación de los pacientes para cada método y están representados en las gráficas 18 y 19 por las barras verdes. La longitud de las barras (rango de error) permiten medir y comparar la flexibilidad entre los clasificadores implementados, siendo mejor entre más pequeña sea dicha longitud, pues en este caso los porcentajes de error tendrán una menor variación lo que representa la adaptabilidad del clasificador a señales procedentes de diferentes pacientes.

➤ Flexibilidad de los clasificadores utilizando la técnica DT-DWT

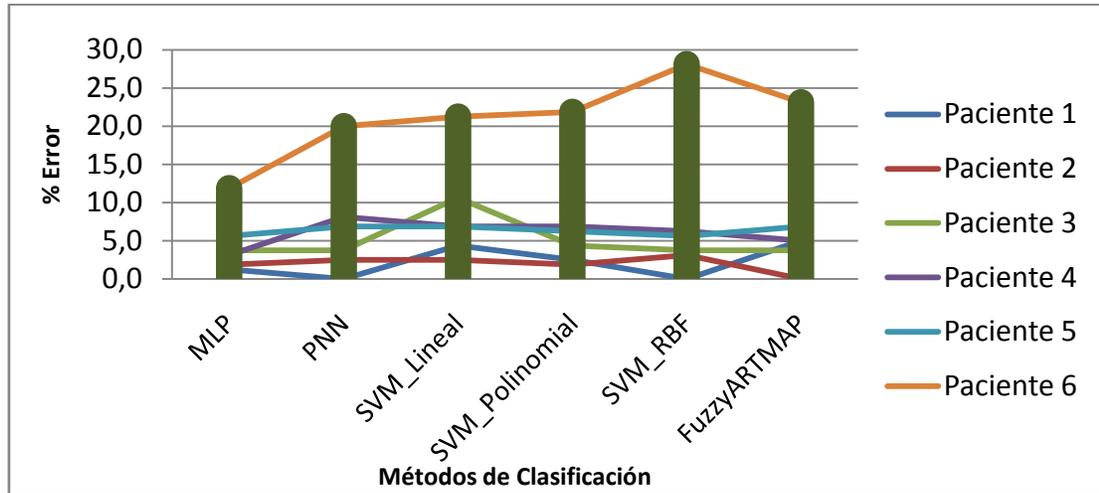


Grafica 18. Comparación de la flexibilidad de los clasificadores con la técnica DT_DWT

	MLP	PNN	SVM_Lineal	SVM_Polinomial	SVM_RBF	FuzzyARTMAP
Rango de error	10,625	21,875	25,625	20,625	20	22,5

Tabla 18. Comparación de la flexibilidad de los clasificadores con la técnica DT_DWT (valores en %).

➤ **Flexibilidad de los clasificadores utilizando la técnica DT-WPT**



Gráfica 19. Comparación de la flexibilidad de los clasificadores con la técnica DT_WPT

	MLP	PNN	SVM_Lineal	SVM_Polinomial	SVM_RBF	FuzzyARTMAP
Rango de error	10,625	20	18,75	20	28,125	23,125

Tabla 19. Comparación de la flexibilidad de los clasificadores con la técnica DT_WPT (valores en %).

El clasificador MLP se presenta como el más flexible con el menor rango de error. Otro aspecto que se puede tener en cuenta para calificar este método como el más flexible es que fue capaz de proporcionar a la clasificación de las señales emitidas por el paciente 6 un porcentaje de error pequeño frente a los obtenidos con los otros clasificadores. El método RBF es el menos flexible, pues presenta el rango de error más amplio al generar mayor porcentaje de error con el paciente seis.

5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

- El uso de vectores híbridos genera buenos resultados en el proceso de clasificación, ya que ellos aportan más información acerca de las características de la señal que en conjunto marcan más las diferencias entre las señales pertenecientes a diferentes movimientos.
- La eficiencia y el buen desempeño de los clasificadores de señales EMG depende en gran medida de la extracción de características y la reducción de dimensionalidad, ya que estas son fases que simplifican el proceso al extraer la información más relevante de una determinada señal.
- Existen diversas técnicas de clasificación de patrones, sin embargo las técnicas inteligentes, por su capacidad de aprendizaje, son una mejor alternativa para la clasificación de señales EMG.
- La selección de los parámetros del clasificador debe hacerse en lo posible para cada paciente, pues las señales EMG difieren por causas como la ubicación de los electrodos, la disposición y la fisiología de cada paciente (grosor de la piel, longitud del músculo, etc), entre otros.
- El MLP es el clasificador que se muestra más preciso y flexible de los implementados, debido que no se limita a memorizar los datos, sino que es capaz de generalizar (interpolarse y extrapolarse). A pesar de que el algoritmo de entrenamiento es lento y en muchos casos termina en óptimos locales que son una solución poco satisfactoria, los resultados de clasificación son bastante buenos y generalmente existe la posibilidad de mejorarlos entrenando la red

las veces que sea necesario hasta obtener una aproximación al error deseado. Su más grande desventaja es que hallar la red óptima es un trabajo de prueba y error, lo que genera demoras en el diseño y la duda de si el diseño implementado es con el que mejores resultados se pueden obtener, pues existen infinitas posibles configuraciones.

- Todos los clasificadores utilizados tienen un buen desempeño en cuanto a la precisión, ellos presentan un porcentaje de acierto en un rango del 89% al 96% independientemente de la técnica utilizada. Sin embargo el clasificador más preciso es el MLP, con un porcentaje de acierto de 95,938% con vectores DT-DWT y de 95,417% con DT-WPT. Por otra parte el clasificador SVM-Lineal, es de los que presenta una tasa de acierto más baja (89,375 con DT-DWT y 91,250 DT-WPT) debido a que la clasificación que realiza es de los procesos más simples pues supone una separación lineal de clases.
- Teniendo en cuenta que el tiempo de procesamiento calculado corresponde a la clasificación de 160 muestras, se concluyó que en general, todos los clasificadores tienen un tiempo de respuesta muy rápido. Entre ellos el más veloz es el SVM – RBF y el que presenta más demoras es la red Fuzzy ARTMAP. Aunque los valores sean muy bajos en la clasificación, hay que tener en cuenta que para la construcción de una prótesis, se requiere de muchos mecanismos y programas para el procesamiento de la señal que funcionando en conjunto emplean mucho más tiempo.
- Los resultados muestran que la red MLP es la más precisa y flexible, sin embargo las SVM que utilizan kernels RBF son más veloces en el proceso de clasificación y sus resultados en cuanto a precisión son buenos (aunque no mejores que los obtenidos con la MLP), por lo que al ponderar la exactitud con el tiempo de clasificación, se obtiene que SVM-RBF es más potente, pues

relaciona de mejor manera estos parámetros relevantes en la implementación real.

- Un clasificador además de preciso y rápido debe ser adaptable; ésta es una propiedad que cumple el clasificador MLP. En éste, es posible reducir considerablemente el porcentaje de error de clasificación del paciente seis que es uno de los que presenta valores de error más elevados con respecto a los demás pacientes. Esto conlleva a pensar, que aunque dichas señales tengan posibles problemas de ruido, el MLP es capaz de clasificarlas correctamente.
- La evaluación y elección final de un clasificador depende de la ponderación de los parámetros definidos para la evaluación y lo que se requiera en la implementación real, pues en ocasiones se suele sacrificar el Tiempo de clasificación a favor de la Tasa de Éxito, mientras que en otras circunstancias donde la clasificación debe ser en tiempo real, el tiempo de clasificación tiene prioridad.
- Teniendo como prioridad el tiempo de respuesta de una prótesis de mano, el clasificador SVM con Kernel RBF es la mejor opción para la implementación debido a su bajo tiempo de respuesta y buen porcentaje de precisión.
- Teniendo en cuenta que cada día aumenta el número de personas con limitaciones físicas y amputación de la extremidades superiores, la ejecución de este trabajo de grado hace un aporte al desarrollo de prótesis de mano, pues contribuye con la exploración de algunos métodos ó técnicas inteligentes para la clasificación de las señales EMG y de esta manera identificar el tipo de agarre deseado por un paciente de manera eficiente.

5.2 RECOMENDACIONES PARA TRABAJOS FUTUROS

- Ampliar la investigación sobre el clasificador MLP y alguna manera de disminuir el tiempo de clasificación, pues teniendo en cuenta que sus resultados en cuanto a la precisión son muy buenos, su potencia podría mejorar si disminuye el tiempo de procesamiento
- En el desarrollo de este documento se trabajó con señales EMG estacionarias, queda abierta la posibilidad realizar todo el proceso de reconocimiento de patrones con señales adquiridas en el cerebro, que es el lugar donde se originan los movimientos.

BIBLIOGRAFIA

- [1] J. F. Franco y E. Giraldo, "Reconocimiento de Patrones de Movimiento del Sistema Fisiológico Brazo – Antebrazo a partir de Señales Electromiográficas", Tesis de grado, Universidad Tecnológica de Pereira, Pereira, Colombia, 2004.
- [2] K. Sharman y A. E. Alcázar, "Interfaz Cerebro-Computadora", Revista del Instituto Tecnológico de Informática, Actualidad TIC 2004, Grupo de Sistemas Adaptativos Complejos, Universidad Politécnica de Valencia, Valencia, España, 2004.
- [3] F.D. Farfán, J.C. Politti y C. J. Felice, "Evaluación de Patrones Temporales y Espectrales para el Control Bioeléctrico", XV Congreso Argentino de Bioingeniería, III Jornadas de Ingeniería Clínica, SABI 205, Paraná, Entre Rios, Argentina, 2005.
- [4] A. Andrade y A. B. Soares, "EMG Pattern Recognition for Prosthesis Control", in 16th Brazilian Congress of Mechanical Engineering, Special session on bioengineering, Uberlândia, Brasil, 2001, pp. 14-21.
- [5] D. P. Gasca y L. A. Rojas, "Extracción de Características Descriptoras de Movimientos de La Mano a Partir de Señales Electromiográficas (Emg) Aplicando Técnicas Wavelets", Trabajo de grado, Universidad del Cauca, Popayán, Colombia, 2007.
- [6] S. Du, "Feature Extraction for Classification of Prehensile Electromyography Patterns", Master's thesis, San Diego State University, San Diego, California, 2003.
- [7] M. Zecca, S. Micera, M. C. Carrozza y P. Dario, "Control of Multifunctional Prosthetic Hands by Processing the Electromyographic Signal". Critical Reviews™ in Biomedical Engineering, vol. 30, Italy, 2002, pp. 459–485.
- [8] J. Martínez y R. Castro, "Análisis de la Teoría de Ondículas orientadas a las aplicaciones de ingeniería eléctrica: fundamentos", Universidad Politécnica de Madrid, Departamento de ingeniería eléctrica, Madrid, España 2002.
- [9] V. Wickerhauser, "Adapted Wavelet Analysis: from Theory to Software", Basic wavelet book, A. K. Peters, Ltd., Wellesley, Massachusetts, 1995.
- [10] J. L. Martín, S. Palazuelos, L. Boquete, M. Mazo y D. Provencio, "Estudio de la Transformada de Fourier y la Transformada Wavelet como herramienta de Análisis y Clasificación de Señales EEG", en XX Congreso Anual de la Sociedad Española de Ingeniería Biomédica, Zaragoza, 2002, pp. 309-313.
- [11] R. Carreño y M.I. Vuskovic, "Wavelet Transform Moments For Feature Extraction From Temporal Signals", en 2nd International Conference on Informatics in control, Automation and Robotics (CINCO), 2005, Barcelona, Spain, 2005.
- [12] O.E. Gualdrón, "Desarrollo de diferentes métodos de selección de variables para sistemas multisensoriales", Tesis Doctoral, Universidad de Rovira y Virgili, Escuela Técnica Superior de Ingeniería, Departamento de Ingeniería, Electrónica, Eléctrica y Automática, Tarragona España, 2006.

- [13] A. Guerra, "Aprendizaje Automático: Clasificación", Tesis de Maestría en Inteligencia Artificial, Universidad Veracruzana, Facultad de Física e Inteligencia Artificial, México, 2004.
- [14] F. Van Der Heijden, R. Duin, D. de Ridder y D. Tax, "Classification, Parameter Estimation and State Estimation, An Engineering Approach using MATLAB", University of Twente, Delft University of Technology, The Netherlands, Editorial Jhon Wiley & Sons, Ltd., 2006, pp. 142, 149.
- [15] J. F Sigut, "Integración de técnicas de análisis y clasificación de datos mediante un sistema basado en el conocimiento para el diagnóstico de patologías cerebrales y dislexias", Tesis Doctoral, Universidad de la Laguna, Tenerife, Canarias, 2002.
- [16] E. Aldabas, "Introducción al reconocimiento de patrones mediante redes neuronales", en Jornadas de conferencias de ingeniería electrónica 2002, Universidad Politécnica de Cataluña, Campus de Terrassa, Terrassa, Barcelona, 2002.
- [17] S. Haykin, "Neural Network. A comprehensive Foundation", IEEE Press 1994, pp. 138.
- [18] M. A. Acosta, H. Salazar y C.A. Zuluaga, "Tutotial de redes neuronales", Universidad Tecnológica de Pereira, Pereira, Colombia, 2002.
- [19] K.M. Peshan, P.W. Jayathilake, R. Ramanan, S. Fernando, Suthrjan, Dr. Chatura De Silva, "Speech Recognition Using Neural Network", Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka, 2000.
- [20] S. Valero, M. Ortiz, C. Senabre, M. Peñarrubia, A. Gabaldón y F. Garcia, "Clasificación de consumidores eléctricos mediante el uso de varias Técnicas de redes neuronales artificiales, e identificación de nuevos clientes con las redes entrenadas", en "9 Congreso Hispano-Luso de Ingeniería Eléctrica, España, 2005.
- [21] G. Colmenares. "Redes Neuronales, Función de Base Radial (RBF)", 2007. Disponible en: http://www.webdelprofesor.ula.ve/economia/gcolmen/programa/redes_neuronales/capitulo4_funciones_bases_radiales.pdf.
- [22] M. Martínez y J. Martínez, "Clasificación y Localización de Faltas, utilizando Wavelets y Redes Neuronales", Noveno Congreso Hispano - Luso de Ingeniería Electrica. Marbella. España. Junio-Julio 2005.
- [23] G. T. Albanis y R. A. Batchelor, "Using Probabilistic Neural Networks and Rule Induction Techniques to Predict Long-Term Bond Ratings", in Proceedings of the 5th Annual Conference on Information Systems Análisis and Synthesis, Orlando, US, 1998.
- [24] A. Hadad, D. Evin, B. Drozdowicz y O. Chiotti, "Integración de Modelos de Seguimiento Temporal en el Monitoreo de Pacientes Críticos", en 36 Jornadas Argentinas de Informática e Investigación Operativa (JAIO), 2007, Mar del Plata.
- [25] M. Pardo, G. Sberveglieri, "Classification of electronic nose data with support vector machines", Sensors and Actuators B: Chemical, Volume 107, Issue 2, 29 June 2005, pp. 730-737.
- [26] P. Lorca, J. De Andrés, J. Díez, J.J. Del Coz y A. Bahamonde, "El análisis de preferencias: Un nuevo enfoque para el estudio de la rentabilidad empresarial", Investigaciones Económicas, vol. XXXI (2): 221-262, 2007.

- [27] C. Burges, "A tutorial on Support Vector Machines for pattern recognition", Bell Laboratories, Lucent Technologies, Boston, 1998.
- [28] H. Kuhn y A. Tucker, "Nonlinear programming", in proceedings 2nd Berkeley Symposium on Mathematical Statistics and probabilistic, University of California Press, Berkeley, California, 1951, pp. 481-492.
- [29] K. Bennet y O. Mangasarian, "Robust linear programming discrimination of two linearly separable set", Optimizations methods and Software, Vol. 1, 1992, pp. 23 - 34.
- [30] C. Cortes y V. Vapnik, "Support vector network", Machine Learning, Vol.20, 1995, pp. 273 – 297.
- [31] C. Seijas, A. Caralli, S. Villasana, L. Sáenz y F. Arteaga. "Predicción de hipertensión arterial usando máquinas de vectores de soporte". Revista Ingeniería Universidad de Carabobo. vol. 13, Venezuela, 2006.
- [32] J. Weston y C. Watkins. "Support Vector Machines for Multi-Class Pattern Recognition". in Proceedings of the Seventh European Symposium On Artificial Neural Networks, Europa, 1999.
- [33] C. M. Duran Acevedo. "Diseño y optimización de los subsistemas de un sistema de olfato electrónico para aplicaciones agroalimentarias e industriales", Tesis Doctoral, Escola Tècnica Superior D' Enginyeria, Universitat Rovira I Virgili, Tarragona, España, 2005.
- [34] B. Acha Piñero. "Segmentación y clasificación de imágenes en color. aplicación al diagnóstico de quemaduras", Tesis doctoral, Área de Teoría de la Señal y Comunicaciones, Departamento de Ingeniería Electrónica, Escuela Superior de Ingenieros, Universidad de Sevilla, 2002.
- [35] I. A. Manolis. "A Brief Description of the Levenberg-Marquardt Algorithm Implemented by levmar", Foundation for Research and Technology, Institute of Computer Science, Greece, 2005.