



Universidad Autónoma de San Luis Potosí

Facultad de Ingeniería

Centro de Investigación y Estudios de Posgrado

**Sistema para Diagnóstico de Enfermedades Neuromusculares
mediante procesamiento de señales Mioeléctricas**

T E S I S

Que para obtener el grado de:

Maestro en Ingeniería de la Computación

Presenta:

Cristina Santillán Martínez

Asesor:

Dr. Carlos Soubervielle Montalvo

San Luis Potosí, S. L. P.

Octubre de 2020





UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ
FACULTAD DE INGENIERÍA

Centro de Investigación y Estudios de Posgrado
Posgrado en Maestría en Ingeniería en Computación

El firmante, Dr. Carlos Soubervielle Montalvo, profesor-investigador de la Universidad Autónoma de San Luis Potosí.

HACE CONSTAR

Que la tesis que tiene por título “Sistema para Diagnóstico de Enfermedades Neuromusculares mediante procesamiento de señales Mieléctricas”

Presentada por la L.I. Cristina Santillán Martínez

Para optar al grado de:

Maestra en Ingeniería de la Computación

Ha sido realizada bajo mi dirección.

San Luis Potosí, S.L.P. a 8 de Octubre de 2020

Firma

Dr. Carlos Soubervielle Montalvo



UASLP
Universidad Autónoma
de San Luis Potosí



FACULTAD DE
INGENIERÍA



CENTRO DE
**INVESTIGACIÓN
Y ESTUDIOS
DE POSGRADO**

UNIVERSIDAD AUTÓNOMA DE SAN LUIS POTOSÍ
FACULTAD DE INGENIERÍA
Área de Investigación y Estudios de Posgrado

Aclaración

El presente trabajo que lleva por título:

Sistema para Diagnóstico de Enfermedades Neuromusculares mediante procesamiento de señales Mioeléctricas

se realizó en el periodo de Enero de 2012 a 30 de Junio de 2020 bajo la dirección del Dr. Carlos Soubervielle Montalvo.

Originalidad

Por este medio aseguro que he realizado este documento de tesis para fines académicos sin ayuda indebida de terceros y sin utilizar otros medios más que los indicados.

Las referencias e información tomadas directa o indirectamente de otras fuentes se han definido en el texto como tales y se ha dado el debido crédito a las mismas.

El autor exime a la UASLP de las opiniones vertidas en este documento y asume la responsabilidad total del mismo.

Este documento no ha sido sometido como tesis a ninguna otra institución nacional o internacional en forma parcial o total.

Se autoriza a la UASLP para que divulgue este documento de Tesis para fines académicos.

Nombre y Firma del autor

Cristina Santillán Martínez

Universidad Autónoma de San Luis Potosí
Facultad de Ingeniería
Centro de Investigación y Estudios de Posgrado
Maestría en Ingeniería de la Computación

**Sistema para Diagnóstico de Enfermedades Neuromusculares
mediante procesamiento de señales Mieléctricas**

Presenta:

L.I. Cristina Santillán Martínez

Asesor:

Dr. Carlos Soubervielle Montalvo

Octubre 2020

Dedicatoria y Agradecimientos

A lo largo del desarrollo de este documento de Tesis y la culminación de mis estudios de maestría debo destacar que tuve siempre el apoyo de mucha gente.

Mi esposo Alberto, gracias por motivarme a ser mejor, por tu paciencia, apoyo y por tanto más.

A mis padres y hermanos por su apoyo incondicional.

Dr. Carlos Soubervielle Montalvo, gracias por su paciencia, por creer en los alumnos y su apoyo infinito.

Dr. Juan Carlos Cuevas Tello le agradezco asesorarme cuando poco tiempo tenía, agradezco su inmensa calidad humana que hacía sentirme como si nunca me hubiera ido.

Dr. César Augusto Puente Montejano y Dr. Marco Tulio Ramírez Torres, fue un honor tenerlos como sinodales.

Gracias al Posgrado en Computación por aceptar mi permanencia en el posgrado y por todo el apoyo recibido, así como a la Dra. Alejandra Guadalupe Silva Trujillo.

Dr. Juan Antonio Cárdenas Galindo sin su apoyo laboral, mi permanencia en el posgrado no se hubiera logrado.

Dra. Liliana Félix Ávila, no tengo palabras para agradecerle todo su apoyo, usted me dio la mano sin conocerme y me dio la confianza para salir adelante.

Dr. Víctor Manuel Cárdenas Galindo le agradezco por la beca para concluir mi posgrado.

A Dora, nuestra compañera de casa, amiga y muchas veces confidente. Gracias por tu apoyo, amistad, compañerismo y demostrarnos la hermandad entre posgrados. Priscila, gracias por tu apoyo y compañerismo.

A todas las personas que me apoyaron a la distancia en el CIEP desde el personal administrativo hasta el Jefe de Investigación y Posgrado, y al Dr. Ricardo Romero Méndez, mi más sincero agradecimiento.

Resumen

La electromiografía (EMG) tanto superficial como invasiva son herramientas cada vez más utilizadas para medir las corrientes eléctricas generadas por los músculos durante su contracción y se representa como actividad muscular. Las señales EMG complicadas, porque las controla el sistema nervioso central. El análisis cuantitativo en la electromiografía clínica es de alto valor, pues permite una evaluación estandarizada y específica de los hallazgos neurofisiológicos pues se evalúa los trastornos neurológicos.

Un problema de salud en México son las enfermedades neuromusculares, debido al incremento del estrés en las personas, sedentarismo, obesidad y/o diabetes, así como la desnutrición o alteraciones digestivas. Por otro lado, actualmente los servicios de salud públicos en México no son tan eficaces para el diagnóstico y tratamiento de estas enfermedades.

El presente trabajo de Tesis se presenta un sistema para el diagnóstico de enfermedades neuromusculares, es decir, es un apoyo para el diagnóstico de miopatías y neuropatías mediante la metodología de análisis, clasificación de las señales para la obtención y procesamiento de señales Electromiográficas (EMG). La adquisición de señales electromiográficas se hace con un par de electrodos superficiales, éstos van conectados a una tarjeta electromiográfica. A través de este dispositivo se guardan las señales EMG en una base de datos, se adquieren las características de cada una de las señales. Las características extraídas se ingresan a un módulo clasificador mediante Redes Neuronales Artificiales el cual determina si los datos ingresados son de pacientes sanos o no.

Se plantea un estudio EMG del sistema nervioso periférico central de miembros inferiores y superiores en las cuales se determinan las ubicaciones de la colocación de los electrodos y se obtenga una amplitud mayor de las señales.

Para lograr una precisión mayor en el prediagnóstico mediante señales EMG se implementó una base de datos de señales EMG de pacientes con Miopatía y Neuropatía recabada a través de diferentes repositorios, investigadores y biotecnólogos con datos

precisos de pacientes con algún padecimiento en específico para darle formalidad al trabajo de tesis.

Los resultados del sistema desarrollado demuestran que entre mayores datos se le ingresen a la red neuronal, más precisa será, para este experimento el porcentaje de aceptación es del 98%.

Palabras clave: EMG, Base de Datos, diagnóstico, neuropatía, miopatía, reconocimiento de patrones, clasificación, redes neuronales artificiales.

Índice general

Dedicatoria y Agradecimientos	i
Resumen	ii
Índice general	v
Índice de Figuras	ix
Índice de Tablas	xii
Índice de Ecuaciones.....	xiii
Introducción	1
Planteamiento del problema	3
Objetivo General	5
Objetivos Específicos	6
Justificación.....	6
1. Fundamentos de electromiografía.....	7
1.1 Electromiografía	8
1.1.1 Propagación y potencial de acción	9
1.2 Técnicas de sensado.....	11
1.2.1 Electroodos	11
1.3 Estado del arte relacionado al análisis de señales electromiográficas	16
1.3.1 Posicionamiento de electrodos superficiales	20
2. Metodología para Desarrollo del Sistema	23
2.1 Sistema neuromuscular	24
2.2 Diseño	26
2.3 Población.....	28
2.4 Tipo de muestreo.....	28
2.5 Técnicas de procesamiento de señales	28
2.5.1 Registro de la actividad electromiográfica	29
2.5.2 Procesamiento de señales electromiográficas.....	31
2.6 Extracción de características	32
2.6.1 Valor absoluto medio mejorado (EMAV)	32
2.6.2 Longitud de onda mejorada (EWL)	33
2.6.3 Valor absoluto medio (MAV)	33

2.6.4	Longitud de forma de onda (WL)	34
2.6.5	Cruce cero (ZC).....	34
2.6.6	Cambio de señal de pendiente (SSC).....	35
2.6.7	Media cuadrática (RMS)	35
2.6.8	Cambio de amplitud promedio (DAMV).....	36
2.6.9	Diferencia valor de desviación estándar absoluto (DASDV).....	36
2.6.10	Detector de registro (LD)	37
2.6.11	Valor absoluto medio modificado 1(MMAV1)	37
2.6.12	Valor absoluto medio modificado 2 (MMAV2)	38
2.6.13	Porcentaje de miopulso (MYOP).....	38
2.6.14	Integral cuadrada simple (SSI)	39
2.6.15	Varianza (VAR).....	39
2.6.16	Amplitud willinson (WA)	39
2.6.17	Longitud máxima del fractal (MFL).....	40
2.6.18	Frecuencia mediana (MDF)	40
2.6.19	Frecuencia media (MNF)	41
2.6.20	Duración	41
2.6.21	Duración del pico (DP).....	42
2.6.22	Amplitud	42
2.6.23	Área.....	43
2.7	Clasificación mediante redes neuronales artificiales	43
2.7.1	Neuronas de entrada.....	44
2.7.2	Neuronas de salida.....	44
2.7.3	Neuronas ocultas.....	44
2.8	Características de los patrones patológicos.....	46
2.9	Diagnóstico de trastornos neuromusculares con señales EMG	47
3.	Técnicas de Reconocimiento de Patrones	49
3.1	Introducción.....	50
3.2	Reconocimiento de patrones	51
3.2.1	Componentes básicos de un sistema de reconocimiento de patrones.....	52
3.2.2	Elementos del reconocimiento de patrones	52
3.3	Módulo clasificador propuesto mediante RNAs.....	54
3.4	Aproximador backpropagation	55

3.4.1	Arquitectura.....	55
3.4.2	Arquitectura de la red neuronal artificial	58
4.	Resultados de los Experimentos.....	61
4.1	Introducción.....	62
4.2	Base de datos	62
4.3	Extracción de características	67
4.4	Clasificador.....	73
4.5	Resultados	94
4.5.1	Resultados bajos.....	95
4.5.1.1	Resultado prueba 86 con 52 muestras.....	95
4.5.1.2	Resultado prueba 87 con 52 muestras.....	97
4.5.1.3	Resultado prueba 89 con 52 muestras.....	99
4.5.1.4	Rsultado prueba 90 con 52 muestras.....	101
4.5.2	Resultados con mejor rendimiento.....	102
4.5.2.1	Resultado prueba 108 con 52 muestras.....	102
4.5.2.2	Resultado con 42 muestras	104
4.5.2.3	Resultado con 74 muestras	106
4.5.2.4	Resultado con 79 muestras	108
4.5.2.5	Mejor resultado con 21 muestras	110
4.5.3	Resultados por grupos	113
	Conclusiones	121
	Trabajo Futuro.....	122
	Referencias	123
	Anexos	130
	Anexo 1. Ubicación de los electrodos.....	130
	Los glúteos.....	131
	Los trapecios y dorsales.....	132
	Las piernas y los gemelos.....	132
	Los abdominales	132
	Pectorales.....	133
	Anexo 2. Planificación del electro diagnóstico	134
	Anexo 3. Código fuente Matlab	135
	Filtro notch	135

Procesamiento de la señal	135
Obtención de señales en archivos con extension .txt	138
Normalización de la señal	138
Cálculo del RMS	139
Cálculo de la frecuencia mediana	141
Extracción de características.....	146
Filtrado	147
Graficar señales	149
Módulo RMS de la extracción de características	168
Anexo 4. Detalle parámetros de la red neuronal.....	171
Anexo 5. Resultados más significativos de los experimentos	172
Prueba 87	172
Pruebas 80% entrenamiento, 10% validación, 10% pruebas con 52 muestras	173
Pruebas 70% entrenamiento, 20% validación, 10% pruebas con 52 muestras	185
Pruebas 65% entrenamiento, 20% validación, 15% pruebas con 74 muestras	200
Pruebas 65% entrenamiento, 20% validación, 15% pruebas con 74 muestras y 16 neuronas	215
Anexo 6. Señales EMG	233
Adquisición propia.....	233

Índice de Figuras

Figura 1. Diagrama de clasificación de la EMG.....	1
Figura 2. Diagrama de experiencias del paciente (UNAM, 2013).....	3
Figura 3. Ejemplo señal EMG en músculo inervado a una amplitud máxima de 1.1 milivolts.....	8
Figura 4. Electrodo bipolar superficial con alineación de plata.....	12
Figura 5. Electrodo EMG concéntrico desechable (Intermedica, 2009).....	13
Figura 6. Electrodo de aguja EMG desechable (Intermedica, 2009).....	13
Figura 7. Electrodo de aguja hipodérmica desechable (Intermedica, 2009)	14
Figura 8. Vista frontal para posicionamiento de electrodos de aguja y superficie (Institut, 2011)	20
Figura 9. Vista dorsal del posicionamiento anatómico de los sitios seleccionados tanto de electrodo de aguja como de electrodo de superficie (Institut, 2011).....	21
Figura 10. Clasificación del sistema muscular.....	24
Figura 11. Diseño general de los módulos	27
Figura 12. Método general para procesar la señal EMG	28
Figura 13. Fisiología de una motoneurona (Raez, Mohd, & Faisal, 2006)	29
Figura 14. Potencial de la unidad motora (Cocero & Recuero, 1971).....	30
Figura 15. Características de la neurona (Foundation, 2020).....	44
Figura 16. Definición de la red neuronal.....	45
Figura 17. Módulo general metodología de diagnóstico	45
Figura 18. Enfoque sistemático al reconocimiento de patrones	52
Figura 19. Sistema de reconocimiento de patrones	54
Figura 20. Arquitectura	58
Figura 21. Señal EMG con miopatía en un tiempo de adquisición de 12 segundos	63
Figura 22. Señal EMG con neuropatía en un tiempo de 10 segundos	64
Figura 23. Señal EMG sana en un tiempo de 5 segundos	64
Figura 24. Señal EMG sana filtrada con filtros pasa banda.....	66
Figura 25. Señal EMG con miopatía filtrada.....	66
Figura 26. Señal EMG con neuropatía filtrada	67
Figura 27. Resultado del experimento número 86.....	96
Figura 28. Resultados del experimento número 87	98

Figura 29. Resultado del experimento número 89.....	100
Figura 30. Resultados del experimento número 90.....	102
Figura 31. Gráfica de desempeño del experimento número 108.....	103
Figura 32. Gráfica del estado de entrenamiento del experimento número 108.....	103
Figura 33. Gráfica de regresión del experimento número 108.....	104
Figura 34. Gráfica de evaluación del desempeño de la Tabla 21.....	105
Figura 35. Gráfica del estado de entrenamiento de la Tabla 21.....	105
Figura 36. Gráfica de regresión del entrenamiento de la Tabla 21.....	106
Figura 37. Resultados del entrenamiento de la Tabla 19.....	107
Figura 38. Histograma de error con 74 muestras.....	107
Figura 39. Gráfica de regresión del entrenamiento de la Tabla 19.....	108
Figura 40. Gráfica de evaluación del desempeño de la Tabla 18.....	109
Figura 41. Gráfica del estado de entrenamiento de la Tabla 18.....	109
Figura 42. Gráfica de regresión del entrenamiento de la Tabla 21.....	110
Figura 43. Gráfica de evaluación del desempeño de la Tabla 23.....	111
Figura 44. Gráfica del estado de entrenamiento de la Tabla 23.....	111
Figura 45. Gráfica de regresión del entrenamiento de la Tabla 23.....	112
Figura 46. Comparación salida deseada contra la resultante y el porcentaje de precisión del experimento de la Tabla 23.....	112
Figura 47. Resultados del experimento del grupo 1, que visualiza las pruebas 92,14,41,32,35,30,37,42,91 y 65 en orden ascendente.....	113
Figura 48. Resultados del experimento del grupo 2, que visualiza las pruebas 54, 52, 60, 2, 64,15,34,38 y 36 en orden ascendente.....	114
Figura 49. Resultados del experimento del grupo 3 que visualiza las pruebas 40,33,43,63,26,48,94,86,97 y 93 en orden ascendente.....	114
Figura 50. Resultados del experimento del grupo 4, que visualiza las pruebas 39,95,3,96,98,5,90,46,85 y 45 en orden ascendente.....	115
Figura 51. Resultados del experimento del grupo 5, que visualiza las pruebas 87,89,49,4,31,88,47,44,53 y 18 en orden ascendente.....	115
Figura 52. Resultados del experimento del grupo 6, que visualiza las pruebas 6,17,51,80,50,67,55,22,23,59,57,29,11,20,24,69,84,56,12 y 99 en orden ascendente.....	116

Figura 53. Resultados del experimento del grupo 7, que visualiza las pruebas 66,16,8,21,58,25,62,10,19,27,28,68,70,71,72,73,74,75,76 y 77 en orden ascendente.....	116
Figura 54. Resultados del experimento del grupo 8, que visualiza las pruebas 78,79,81,82,83,13,9,7,61,121,105,166,141,102,124,118,115,123 y 126 en orden ascendente.....	117
Figura 55. Resultados del experimento del grupo 9, que visualiza las pruebas 127,130,164,139,106,112, 100,119,125,122,157,152,156,149,128,104,140,113,101 y 126 en orden ascendente.....	117
Figura 56. Resultados del experimento del grupo 10, que visualiza las pruebas 168,153,107,129,132,138,159,160,135,165,150,151,144,111,114,108 y 147	118
Figura 57. Figura comparación precisión de las pruebas más bajas 92,14,41,32,35, las pruebas medias 67,55,22,23,59, y las pruebas óptimas 111,114,108 y 147	119
Figura 58. Figura comparación MSE de las pruebas 92, 67 y 111.	120

Índice de Tablas

Tabla 1. Encuesta frecuencia de atención por especialidad (INEGI, 2015)	4
Tabla 2. Rangos de los bioseñales más comunes	9
Tabla 3. Enfermedades afectadas por el potencial de acción motora (SEN, 2004)	11
Tabla 4. Características de la señal miopática y neuropática. (Osuna , González, & Ilzarbe, 2012).....	46
Tabla 5. Enfoque de las técnicas de reconocimiento de patrones y de aprendizaje automático	50
Tabla 6. Funciones de activación comunes para la red backpropagation.....	56
Tabla 7. Diferentes tipos de señales EMG de la base de datos	63
Tabla 8. Parámetros de muestreo recomendados para el acondicionamiento digital de las señales EMG de artículos de revistas seleccionados	65
Tabla 9. Extracción de características de 8 sujetos con miopatía	68
Tabla 10. Extracción de características de 7 sujetos con neuropatía	69
Tabla 11. Extracción de características de 19 sujetos sanos	70
Tabla 12. Extracción de características de 44 sujetos con miopatía y neuropatía (ambos diagnósticos).....	72
Tabla 13. Tipo de redes neuronales.....	73
Tabla 14. Funciones de entrenamiento	75
Tabla 15. Funciones de aprendizaje de adaptación	75
Tabla 16. Función de transferencia.....	75
Tabla 17. Pruebas que se desarrollaron para esta tesis combinando los parámetros de las RNAs disponibles	80
Tabla 18. Clasificador 1, con datos de entrada en la red neuronal representando a las columnas un total de 79 sujetos para entrenamiento (21 pacientes sanos y 59 pacientes enfermos)	85
Tabla 19. Clasificador 2, con datos de entrada en la red neuronal representando a las columnas un total de 74 sujetos para entrenamiento (14 pacientes sanos y 59 pacientes enfermos)	88
Tabla 20. Clasificador 3, con datos de entrada en la red neuronal representando a las columnas un total de 52 sujetos de entrenamiento (49 enfermos y 3 sanos) ...	90

Tabla 21. Clasificador 4, con datos de entrada en la red neuronal representando a las columnas un total de 42 sujetos para entrenamiento (21 pacientes sanos y 21 pacientes enfermos)	92
Tabla 22. Datos para pruebas en la red neuronal representando a las columnas un total de 6 sujetos para entrenamiento (3 pacientes sanos y 3 pacientes enfermos)	92
Tabla 23. Clasificador 5, con datos de entrada en la red neuronal representando a las columnas un total de 22 sujetos para entrenamiento (11 pacientes sanos y 11 pacientes enfermos)	93
Tabla 24. Parámetros utilizados con mejor rendimiento de las pruebas 108, 111, 114 y 147, así como para 44 y 42 muestras.	118
Tabla 25. Parámetros utilizados con mejor rendimiento de las pruebas bajas 92,14,41,32,35, las pruebas con resultado medio 67,55,22,23,59	120

Índice de Ecuaciones

Ecuación 1. Valor absoluto medio mejorado (EMAV).....	32
Ecuación 2. Longitud de onda mejorada (EWL)	33
Ecuación 3. Valor absoluto medio (MAV).....	33
Ecuación 4. Longitud de forma de onda (WL)	34
Ecuación 5. Cruce cero (ZC).....	34
Ecuación 6. Cambio de señal de pendiente (SSC)	35
Ecuación 7. Media cuadrática (RMS).....	35
Ecuación 8. Cambio de amplitud promedio (DAMV)	36
Ecuación 9. Diferencia valor de desviación estándar absoluto (DASDV)	36
Ecuación 10. Detector de registro (LD)	37
Ecuación 11. Valor absoluto medio modificado 1(MMAV1).....	37
Ecuación 12. Valor absoluto medio modificado 2 (MMAV2).....	38
Ecuación 13. Porcentaje de miopulso (MYOP)	38
Ecuación 14. Integral cuadrada simple (SSI)	39

Ecuación 15. Varianza (VAR)	39
Ecuación 16. Amplitud willinson (WA).....	40
Ecuación 17. Longitud máxima del fractal (MFL)	40
Ecuación 18. Frecuencia mediana (MDF)	41
Ecuación 19. Frecuencia media (MNF)	41
Ecuación 20. Duración.....	42
Ecuación 21. Duración del pico.....	42
Ecuación 22. Amplitud	42
Ecuación 23. Activación de la neurona	56
Ecuación 24. Salidas de la red.....	56
Ecuación 25. Señal de error para la neurona de salida.....	57
Ecuación 26. Error total	57
Ecuación 27. Error cuadrático promedio	57
Ecuación 28. Corrección de los pesos	57
Ecuación 29. Conexiones se modificados.....	58
Ecuación 30. MSE	90

Introducción

La electromiografía (EMG) como definición exacta es el registro y estudio de la actividad eléctrica espontánea y voluntaria de los músculos, ya sean lisos o estriados (IENSA, 2020). De acuerdo con el uso de la EMG, se determina el tipo de electrodo de registro; de manera que la electromiografía de aguja hace distinciones entre las lesiones sistema nervioso central (SNC) o sistema nervioso periférico (SNP), mientras que la electromiografía de superficie (sEMG) tiene aplicaciones de carácter biomecánico, rehabilitación, psicológico, y neurológico.

La sEMG se basa en retroalimentación, dónde el uso de electrodos superficiales es importante para monitorear la actividad muscular. Por medio de esta técnica se descubrió que la actividad eléctrica de un músculo podría disminuir o aumentar su contracción por medio de control consiente. A continuación, se muestra en la Figura 1 la clasificación de la EMG.

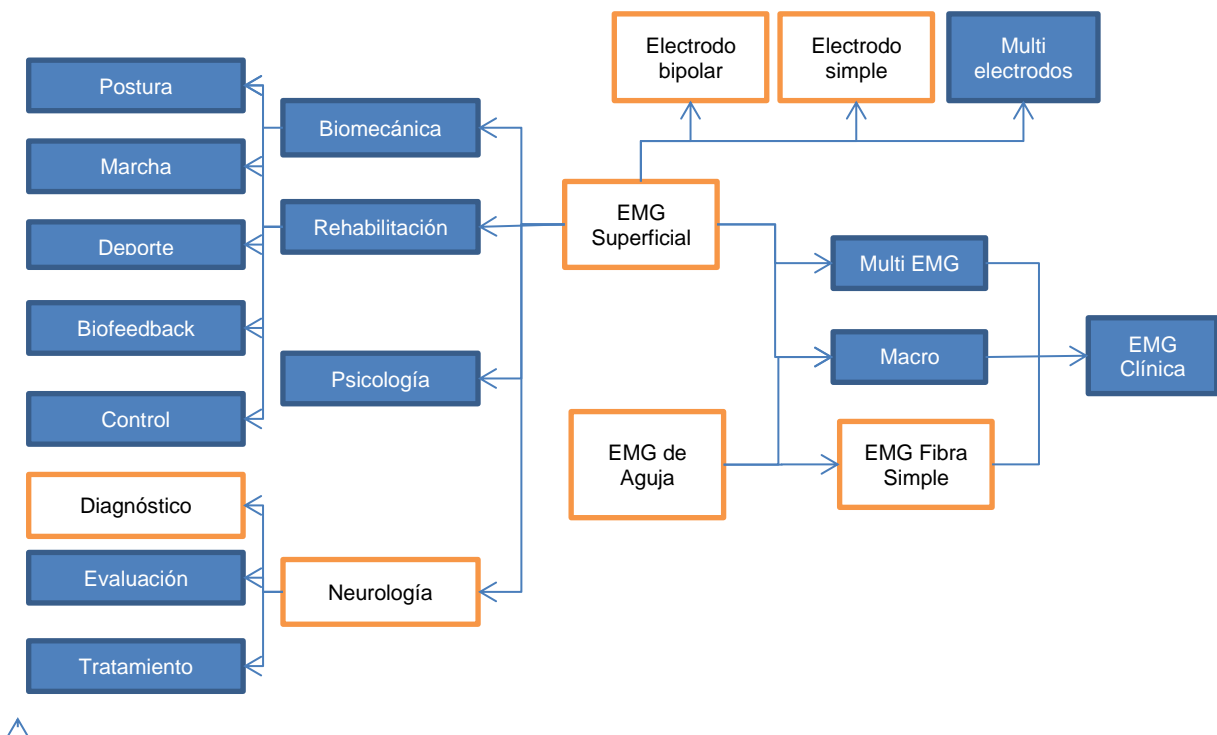


Figura 1. Diagrama de clasificación de la EMG

Para lograr la aplicación del diagnóstico de enfermedades EMG se siguió la ruta de color naranja, a través de aplicaciones en neurología (estudio del sistema nervioso y muscular) con adquisición de señales EMG superficiales y directamente en fibras musculares. La

obtención de señales es a través de electrodos superficiales (bipolares o simples) o de aguja (de fibra simple), para su procesamiento y análisis. Los recuadros a la izquierda en color azul están relacionados con aplicaciones en rehabilitación, biomecánica y psicología.

La señal sEMG muestra el comportamiento muscular y la situación del sistema nervioso que lo determina, razón que demuestra la gran aplicabilidad de la sEMG como técnica para el diagnóstico clínico. La técnica sEMG involucra la aplicación de electrodos de superficie sobre la piel, estos electrodos captan la actividad eléctrica de las fibras musculares y envían información para su procesamiento y su análisis. La técnica involucrada para este concepto es a través de la EMG clínica, utilizando electrodos multicanal para el sistema nervioso periférico, con la finalidad de obtener un prediagnóstico, como se muestra en la Figura 1 en los recuadros color naranja.

Para este proyecto de Tesis el sistema que se desarrollo es para el diagnóstico de enfermedades neuromusculares mediante la toma de la señal EMG. Las señales son almacenadas en una base de datos. Las señales son pre procesadas para eliminar ruidos resultantes de la línea eléctrica, después del preprocesamiento a la señal filtrada se le obtiene 19 características relevantes para el análisis de miopatías, neuropatías y casos sanos. El paso siguiente es ingresar las características obtenidas de las señales, a un clasificador basado en redes neuronales artificiales que permite el diagnóstico de enfermedades neuromusculares.

Con el propósito de automático de la clasificación de señales EMG de clases múltiples, un aprendizaje desarrollado se ha utilizado el algoritmo FFBP. Los resultados experimentales del conjunto de datos EMG muestra el rendimiento mejorado del método propuesto, y es evidente que el método propuesto se puede utilizar eficazmente para el diagnóstico de enfermedades neuromusculares, con una precisión superior al 90% gracias a la utilización de la combinación de diferentes características. Es necesario hacer hincapié en la aportación de este trabajo de tesis, puesto se utilizaron señales de diversas fuentes, que en el capítulo de resultados son detallados, y que las siguientes referencias fueron de gran aporte para lograr este resultado.

Este proyecto de tesis está sustentado en (Pauk, 2008) (Sadi-Ahmed, Messekher, Namane, Kedir-Talha, & Mekaoui, 2017) (González I. J., 2012) (Singh, Dutta, & Travieso, 2017) (Phinyomark, Thongpanja, Hu, Phukpattaranont, & Limsakul, 2013) (Deepika & Wadhvani, 2012) (Ahmad & Ansari, 2012)

Planteamiento del problema

Los sistemas de salud que tiene la República Mexicana tienen un déficit de atención al paciente en tiempo y forma. Según un estudio de satisfacción con la atención médica de la Facultad de Medicina de la Universidad Nacional Autónoma de México edición febrero 2013 los pacientes antes de llegar a ser atendidos por un especialista pasaron por varios filtros (citas a través de internet o con el personal administrativo, cita con un médico general que los valora en un primer momento, estudios de laboratorio y/o imagen). Las calificaciones negativas del servicio médico, por lo general dio como resultado hacia el personal administrativo y las enfermeras; en menores casos, el médico especialista. (Hamui, Fuentes, Aguirre, & Ramírez, 2015)



Figura 2. Diagrama de experiencias del paciente (UNAM, 2013)

Los porcentajes de dos grupos para la atención médica en segundo y tercer nivel de atención: 28% de primera vez y 72% de subsecuente. La subsecuencia denota un estado de salud afectado que requiere frecuentar periódicamente las especialidades. Los datos cuantitativos lo corroboraron: la mitad de los usuarios había acudido más de 11 veces al hospital y el 10% fue atendido en más de tres especialidades.

Porcentaje de frecuencia de atención por especialidad

Número de veces atendido en el hospital		Diferentes números de especialidades en las que fue atendido	
Rango	Porcentaje	Número de especialidades	Porcentaje
Hasta 2 veces	15%	1	46%
De 3 a 5	17%	2	14%
De 6 a 8	9%	3	5%
De 9 a 11	9%	Más de 3	10%
Mas de 11	50%		

Tabla 1. Encuesta frecuencia de atención por especialidad (INEGI, 2015)

De acuerdo con la Tabla 1, en los servicios de salud desde la primera vez que se acude a la clínica hasta su diagnóstico promedian al menos 11 visitas antes de darle un diagnóstico al paciente que en tiempo promedio se identifica de 2 a 6 meses, siendo el paciente él más perjudicado, sin contabilizar las horas de traslado del lugar de origen.

Por lo que se propone un sistema modular no invasivo, capaz de determinar si un paciente tiene una neuropatía, miopatía o está sano en el lapso menor a 1 día, y evitar esos tiempos desgastantes para el paciente y su familia.

El estudio de las señales electromiográficas es de alta complejidad, por lo que en esta primera aproximación como propuesta de este trabajo de tesis es la determinación de algunas patologías de manera general y en un trabajo futuro su comparación con un banco de datos que se está implementado.

El sistema desarrollado capta las señales provenientes de los músculos del paciente por medio de electrodos superficiales localizados en la zona a sensar, mientras que el paciente regula de manera consiente y voluntaria contracciones de los grupos musculares, o relaja los grupos musculares. La base de datos soporta tanto EMG superficial como de aguja con la finalidad de mejorar la clasificación de las señales.

Las señales provenientes de los electrodos de la zona muscular son integradas en el sistema y se guardan en forma de vector, el cual puede ser graficado en tiempo / amplitud. Proporciona los niveles de contracción o relajación, según sea el caso y se guardan en el sistema. Con el sistema es posible determinar si existe alguna deficiencia en la señal.

La localización de los electrodos varía de acuerdo a la posición del individuo. Se tomaron aspectos clave para la posición de los electrodos de superficie, la vista frontal y la vista dorsal.

Las propiedades de las señales EMG tanto en el dominio del tiempo como en la frecuencia dependen de factores tales como: el tiempo y la intensidad de la contracción muscular, así como la distancia entre el electrodo y la zona de actividad muscular.

Las propiedades de la piel (por ejemplo, el espesor de la piel y tejido adiposo). Las propiedades del electrodo y el amplificador y la calidad del contacto entre la piel y el electrodo. Los aspectos más importantes relacionados con la adquisición y el análisis de señales EMG de superficie fueron tratados recientemente en un consenso multinacional llamado SENIAM del inglés (Surface EMG for the non-invasive assessment of muscles), donde se discute desde la construcción del electrodo hasta su ubicación. La medición y la representación de las señales EMG de superficie dependen de las propiedades de los electrodos y su interacción con la piel.

La amplitud EMG tiene una variedad de aplicaciones, tales como la señal de control para prótesis mioeléctricas, estimaciones ergonómicas, sistemas de realimentación (biofeedback), y también ha sido usada para estimar el par asociado a una articulación, y en la última década para diagnóstico.

Objetivo General

Diseñar y evaluar un sistema modular no invasivo capaz de diagnosticar si una persona padece enfermedades neuromusculares mediante el acondicionamiento, análisis y procesamiento de señales EMG.

Objetivos Específicos

- Crear una Base de datos de señales EMG
- Obtener 19 características de las diferentes señales EMG.
- Crear una tabla de características para el análisis de miopatías, neuropatías y casos sanos.
- Crear un clasificador con un porcentaje de aceptación mayor al 95%.
- Diagnóstico de Miopatías y Neuropatías.

Justificación

El censo realizado por el INEGI sobre personas con distintos tipos de discapacidad en México al año 2014 son 6 millones 689 mil 802 personas, lo que representa 5.59% de la población total. (INEGI, 2015)

El sistema que se propone no es invasivo, no es costoso y no necesita de especialistas para su operación.

El impacto en salud pública es múltiple, ya que el paciente no se tiene que trasladar para el prediagnóstico, no tiene que esperar largos períodos para su atención, y lo más importante es que el resultado es inmediato para su pronta atención.

La finalidad del sistema es la detección temprana de enfermedades Neuromusculares para postergar sus efectos en los pacientes y brindarles una mejor calidad de vida.

1. Fundamentos de electromiografía

1.1 Electromiografía

La electromiografía (EMG) es un procedimiento de diagnóstico que se utiliza para evaluar la salud de los músculos y las células nerviosas que los controlan (neuronas motoras). Los resultados de la electromiografía pueden revelar una disfunción nerviosa, una disfunción muscular o problemas con la transmisión de señales de nervios a músculos. (Clinic, 2019) La EMG tiene dos vertientes, EMG superficial y EMG invasiva. Para ver una señal EMG es necesario colocar sensores que nos muestran la señal.

EMG es sinónimo de electromiograma, que es la medición de potenciales eléctricos creados por la contracción de los músculos. Internamente, los músculos generan tensiones alrededor de 100mV cuando se contraen. Estas tensiones son muy atenuadas por el tejido interno y la piel, y que son débiles pero medibles en la superficie de la piel. Típicas señales EMG superficial (sEMG) para los músculos grandes, como el bíceps, están alrededor de 2.1mV en amplitud. Señales sEMG contienen frecuencias que van de 2 Hz o inferior a 500 Hz o mayor. (Alva, 2012)

Como objetivo de este trabajo de tesis es el diagnóstico de miopatías y neuropatías, entendiendo miopatía como una enfermedad muscular y neuropatía como una enfermedad del sistema nervioso, en específico de la no inflamatoria. (RAE, 2019)

Ejemplo de una señal electromiográfica sana

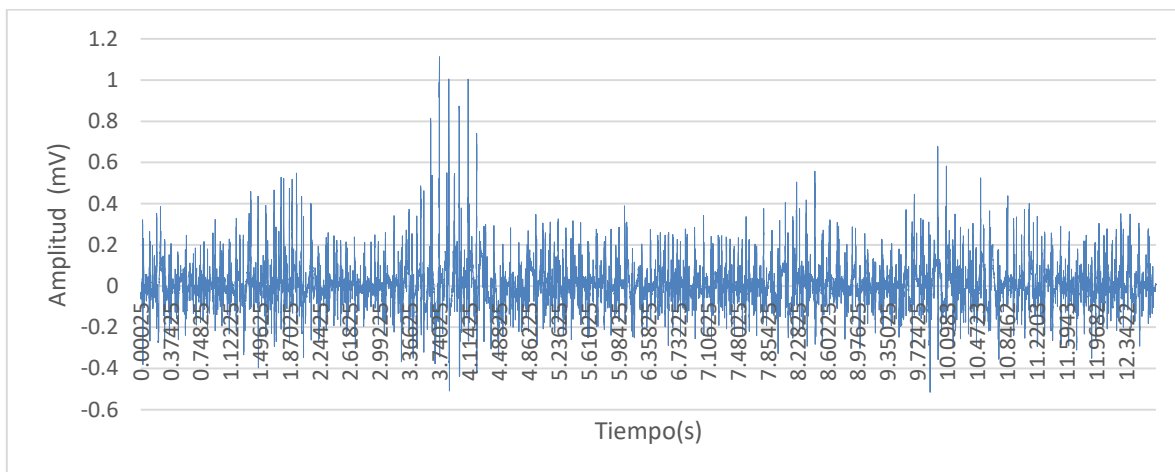


Figura 3. Ejemplo señal EMG en músculo inervado a una amplitud máxima de 1.1 milivolts

La Figura 3 muestra un ejemplo de una señal EMG obtenida de un sujeto sano hombre de 44 años, obtenida del músculo anterior tibial.

1.1.1 Propagación y potencial de acción

1.1.1.1 Bioseñal

Una señal es un medio de transmisión de información, cuya adquisición permite obtener información sobre la fuente que la generó. En el caso de las bio señales, las fuentes son los diferentes sistemas fisiológicos del organismo. La captación de las bio señales permite al médico extraer información sobre el funcionamiento de los diferentes órganos para poder emitir un diagnóstico.

Las medidas médicas pueden agruparse en diversas categorías: biopotenciales, mecánicas, acústicas, imágenes, impedancias, señales biomagnéticas y señales bioquímicas. En el presente capítulo se presentarán las más usuales, como base para comprender mejor la instrumentación y el procesado aplicado sobre las mismas. Los factores más importantes que caracterizan los bioseñales desde el punto de vista de la instrumentación son los rangos de amplitud y frecuencia. Dichos rangos pueden variar en función del método de adquisición, y sus valores son aproximados, considerando tanto normales como anormales de los parámetros listados. (Guerrero, 2010-2011)

Resumen de algunas de los bioseñales más usuales y sus correspondientes rangos.

Bioseñal	Definición	Rango - Amplitud	Rango Frecuencia
Electrocardiograma (ECG)	Actividad eléctrica cardíaca	0.5 – 4 mV	0.01 – 250 Hz
Electroencefalograma (EEG)	Actividad eléctrica cerebral	5 – 300 μ V	DC-150 Hz
Electrogastrograma (EGG)	Actividad eléctrica gástrica	10 μ V-1 μ V	DC- 1 Hz
Electromiograma (EMG)	Actividad eléctrica muscular	0.1 – 5mV	DC – 10 kHz
Electroneurograma (ENG)	Actividad eléctrica nerviosa	0.01 – 3mV	DC – 1k Hz

Tabla 2. Rangos de los bioseñales más comunes

Los rangos mostrados en la Tabla 2 muestra los rangos de las señales eléctricas que se tiene en el cuerpo humano ya sea corazón, cerebro, estómago, músculos y nervios.

1.1.1.2 *El potencial de acción*

En fisiología, el potencial de acción es también llamado un potencial de membrana, el cual es un potencial eléctrico que se forma por la carga diferencial que se forma entre el espacio intracelular y el espacio extracelular. El rango registrado para el potencial de acción aproximadamente es de -70mV y se propaga entre 50 μ V hasta 20 o 30 mV dependiendo del músculo observado. El rango de repetición de una unidad motora es entre 7Hz a 20Hz dependiendo del músculo observado. De acuerdo con lo anterior la suma de la actividad eléctrica se denomina potencial de acción de la unidad motora (MUAP) los cuales son decisivos en la evaluación de una EMG, los resultados de esta evaluación pueden ser debido a enfermedades neuropáticas o enfermedades miopáticas. (Alva, 2012)

1.1.1.2.1 *Enfermedad neuropática*

En este caso el potencial de acción aumenta dos veces de lo normal debido a que las fibras por unidad motora crecen debido a una reinervación de las fibras desnervadas. Los síntomas observables son:

- Incremento en la duración del potencial de acción
- Disminución de las unidades motoras en el músculo (utilizando técnicas de estimación numérica de unidades motoras).

1.1.1.2.2 *Enfermedad miopática*

En este caso existen 3 síntomas visibles, en los que se encuentran:

- Disminución de la duración del potencial de acción.
- Una reducción en el área y la amplitud del radio del potencial de acción.
- Una disminución en el número de unidades motoras en el músculo.

1.1.1.2.3 *Condiciones médicas*

Algunas de las condiciones médicas asociadas con los resultados del potencial de acción son:

Nervio	Músculo
Neuropatía Alcohólica	Miopatía
Neuro miopatía	Poliomielitis
Neuropatía periférica	Polimiositis
Esclerosis lateral amiotrófica	Polineuropatía sensor-motriz
Síndrome del compartimiento anterior	Estenosis espinal
Disfunción del nervio axilar	Disfunción del nervio tibial
Disfunción del nervio femoral	Disfunción del nervio radial
Ataxia de Friedrich	Distrofia muscular de Becker
Síndrome de Guillain-Barré	Plexo Patía Braquial
Síndrome miasténico de Eaton-Lambert	Síndrome del túnel carpiano
Mono neuritis Múltiple	Miopatía centro nuclear
Mononeuropatía	Espondilosis cervical
Enfermedad de la motoneurona	Enfermedad de Charcot-Marie-Tooth
Atrofia del sistema múltiple	Disfunción del nervio peroneo común
Miastenia grave	Denervación
	Dermatomiositis
	Disfunción del nervio mediano distal
	Distrofia muscular de Duchenne

Tabla 3. Enfermedades afectadas por el potencial de acción motora (SEN, 2004)

La Tabla 3 nos muestra algunas enfermedades relacionadas con nervio y músculo, que están directamente relacionadas con miopatías y neuropatías.

1.2 Técnicas de sensado

1.2.1 Electrodo

Se pueden obtener señales EMG para el estudio del movimiento humano utilizando electrodos de superficie o intramusculares generalmente por pares (bipolares). La amplitud y anchura de banda de la señal EMG no están determinadas únicamente por las fuentes electrofisiológicas y sus distancias hasta los electrodos, sino también por los tipos y tamaños de los electrodos utilizados y por esparcimiento entre electrodos. La amplitud y anchura de banda de la señal utilizable van de 10 micro volts a 5 milivolts y de 10 Hz a 10 KHz, respectivamente, dependiendo de la configuración de los electrodos. Sin embargo, incluso para electrodos intramusculares de hilo fino, la mayor parte de la energía de la señal está por debajo de 1 KHz y la amplitud dentro de 2 mV de máximo a mínimo. (Meneses, 2005)

1.2.1.1 Características de los electrodos

Los electrodos superficiales van unidos a la piel sobre el segmento muscular que se está estudiando. Los electrodos de superficie se utilizan para estudiar la actividad de todo el músculo superficial. El espaciamiento entre electrodos determina el volumen de registro o recepción del tejido, resultando los espaciamentos más pequeños en registros más selectivos. Los electrodos de superficie suelen ser de ranura, con pasta de electrodo llenando la cavidad para conseguir más contacto con la piel y reducir la impedancia de los electrodos.

Los electrodos comercializados pueden ser desechables, como los electrodos Electrocardiográficos (ECG), o reutilizables con una protección de plástico y un cuello adhesivo por ambos lados. Su diámetro va de 2 a 10 mm para la parte activa del electrodo.

La Figura 4 muestra los electrodos de cloruro de plata-plata (Ag-Ag Cl) superficiales con pasta de cloruro se utilizan invariablemente debido a sus propiedades de estabilidad y reducción del ruido. (González I. J., 2012)



Figura 4. Electrodos bipolares superficiales con alineación de plata

La Figura 5 muestra los electrodos EMG concéntricos, son portátiles y precisos para realizar mediciones rápidas, como se observa son superficiales y fáciles de colocar.

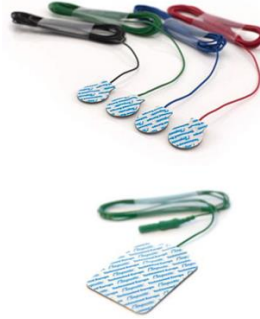


Figura 5. Electrodo EMG concéntricos desechables (Intermedica, 2009)

Los electrodos concéntricos son ideales para mediciones cerebrales, cardíacas y musculares.

1.2.1.1.1 *Electrodos de aguja*

El electrodo de aguja es un alambre muy fino por el cual la corriente eléctrica es adquirida de un bioseñal. El electrodo se introduce al músculo que se desea observar.

Como se muestra en la Figura 6 la forma del electrodo monopolar es una aguja de acero quirúrgico y su revestimiento es politetrafluoroetileno. Por seguridad del paciente, las agujas se desechan después de haber sido utilizadas.



Figura 6. Electrodo de aguja EMG desechable (Intermedica, 2009)

La Figura 7 muestra un ejemplo de aguja hipodérmica, este tipo de agujas son para medir la actividad eléctrica mientras se va introduciendo, esta aguja se podría decir que está hueca para jeringas, las cuales inyectan sustancias que activan o reaccionan el músculo.



Figura 7. Electrodo de aguja hipodérmica desechable (Intermedica, 2009)

1.2.1.2 Recomendaciones europeas con respecto a la detección y acondicionamiento de la señal EMG superficial

Ha habido grandes avances en sEMG en la última década, cuyos resultados aparecen diferentes metodologías salidas de los diferentes laboratorios de investigación clínica. Este hecho lleva a realizar grandes esfuerzos de normalización para que los resultados sean comparables y para crear un gran cuerpo común de conocimientos sobre el uso de la sEMG en sus múltiples ámbitos de aplicación. Por esta razón aparece en 1996 una asociación cuyo nombre es: Electromiografía de superficie no invasiva para la Evaluación de músculos o del inglés Surface Electromyography for Noninvasive Assessment of Muscles (SENIAM), con el objetivo principal de crear un consenso sobre los principales elementos (sensores, la colocación de sensores, procesamiento de señales, y modelado) para permitir el intercambio de datos y resultados obtenidos con la SEMG.

Sensor: se define como el conjunto de electrodos, integrados a un preamplificador, siendo una configuración de dos electrodos. El sensor más frecuentemente utilizado para registrar la señal de sEMG, debe poseer las siguientes propiedades o características:

- Tamaño y forma específica del electrodo.
- Material adecuado.
- Distancia entre electrodos (IED).
- Las modalidades de construcción.

1.2.1.3 *Distancia entre electrodos*

La SENIAM define la distancia entre electrodos como “la distancia entre centros de las áreas de conductividad de los electrodos”. Con respecto a la distancia entre electrodos, la normatividad recomienda que:

- Los electrodos bipolares EMG de superficie tengan una distancia entre electrodos de entre 20mm y 30mm.
- Cuando los electrodos bipolares están siendo aplicados sobre músculos relativamente pequeños, la distancia entre electrodos no debe superar 1/4 de la longitud de la fibra muscular. De esta forma se evitan los efectos debidos a tendones y terminaciones de las fibras musculares. (Meneses, 2005)

1.2.1.4 *Posicionamiento de los electrodos*

Las señales EMG dan una muestra de la actividad eléctrica en los músculos durante una contracción. Sin embargo, estas señales están altamente relacionadas con la posición del electrodo sobre el músculo de interés. Debido a esto, es necesario que la ubicación de los electrodos sea consistente en sesiones consecutivas de estudio y sobre diferentes pacientes. Para determinar la ubicación de los electrodos es recomendado utilizar la normatividad correspondiente donde se encuentran sugerencias para la ubicación de los electrodos sobre 27 zonas musculares distintas. El objetivo al ubicar los electrodos es conseguir una ubicación estable donde se pueda obtener una buena señal electromiográfica. Los electrodos se pueden ubicar sobre la superficie de la piel de manera longitudinal, o transversal.

- Longitudinal: la recomendación es ubicar el electrodo bipolar en la zona media del músculo, esto es, entre la terminación de la neurona motora que envía el impulso eléctrico al músculo (aproximadamente línea media del músculo) y el tendón distal.
- Transversal: la recomendación es ubicar el electrodo bipolar sobre la zona media del músculo, de tal forma que la línea que une los electrodos sea paralela con el eje longitudinal del músculo. (Meneses, 2005)

1.3 Estado del arte relacionado al análisis de señales electromiográficas

Se analizaron diferentes trabajos relacionados que a continuación se detallan brevemente:

A comparative study of EMG and MMG signals for practical applications (Irfan, Sudharsan, Santhanakrishnan, & B. , 2011)

Este artículo se basa en señales EMG para prótesis humanas con un hardware BIOPAC MP SYSTEM y lo compara con la adquisición de señales con LabVIEW. Este artículo fue de gran ayuda para definir con que quería trabajar para procesar las señales. Se comenzó inicialmente trabajando con LabVIEW para el Desarrollo de este Proyecto de tesis, se adquirieron señales de aproximadamente 50 individuos en tiempo real con este software, los cuales sirvieron de pruebas.

A review of EMG recording technique. International journal of engineering science and technology. (Ahmad & Ansari, 2012)

Este estudio investiga la actividad eléctrica de los músculos y la interrelación de los músculos y las fibras nerviosas. Se observa que existe una relación alterna entre las contracciones de bíceps y tríceps; la relajación, fuerza y la contracción. Las señales EMG se han registrado en varias posiciones, como el punto Bíceps, el punto tríceps y las contracciones puntuales. Este artículo fue de gran ayuda puesto que muestra la localización de los electrodos y la instrumentación utilizada para dicho experimento.

Analysis of different EMG signals by segmentation, classification and feature extraction phase (Deepika & Wadhwani, 2012)

Este artículo investiga y experimenta con la actividad muscular de los autores para desarrollar un brazo robótico y automatizar mediante la señal EMG recibida utilizando DWT y clasificadores fuzzy logic.

Application of wavelet analysis in EMG feature extraction for pattern classification (Angkoon Phinyomark, 2011)

En este artículo analizan la extracción de la señal EMG para la clasificación de patrones mediante análisis wavelet para la descomposición de la señal. Los experimentos que

hicieron son con 6 movimientos del brazo: (a) flexión de la muñeca (b) muñeca extensión (c) mano cerrada (d) mano abierta (e) pronación del antebrazo (f) supinación del antebrazo. La descomposición se hizo a través de extracciones más populares como mean absolute value (MAV) y root mean square (RMS). Este artículo fue valioso pues se observó que se utilizaron dos parámetros importantes que actualmente se usan en este proyecto de tesis.

Analysis of EMG signals for automated diagnosis of myopathy. (Singh, Dutta, & Travieso, 2017)

Este artículo está basado en extraer las características estadísticas básicas (Mean of Signal Amplitude, STD, Max Value of signal amplitude, peak to peak value y RMS) de las señales EMG y discrimina entre una señal normal o una señal con miopatía, logra un 87% de precisión y una sensibilidad del 90%. Toma en cuenta también una base de datos EMG para el diagnóstico. Los resultados de su experimento son motivadores para este trabajo de tesis.

The usefulness of mean and median frequencies in electromyography analysis. (Phinyomark, Thongpanja, Hu, Phukpattaranont, & Limsakul, 2013)

Este artículo explora y analiza Mean frequency (MNF) y median frequency (MDF) como medidas en el dominio de la frecuencia que para determinar la fatiga de un músculo usando señales EMG. Las medidas tomadas en este artículo debido a su éxito en éste y en otros experimentos se tomaron en cuenta para este trabajo de tesis.

Tesis: Extracción de características y clasificación de señales electromiográficas utilizando la transformada Hilbert-Huang y redes neuronales por Ing. Irving Aaron Cifuentes González. (Cifuentes, 2012)

Esta tesis se presenta dos métodos basados en Redes Neuronales, el primer método se basa en el reconocimiento de patrones en las señales EMG para identificación de 3 movimientos con la pierna y el otro método realiza una estimación de posición y velocidad angular. En este trabajo se validan algunos datos estadísticos que se usan en el presente trabajo de tesis.

Tesis: Reconocimiento de señales mioeléctricas mediante redes neuronales artificiales
(González I. J., 2012)

Crea una metodología para la obtención de señales superficiales EMG para el reconocimiento de 3 movimientos del brazo a través de Redes Neuronales Artificiales. El presente trabajo de tesis es la continuación de la tesis de Juan Carlos González Ibarra pues demostramos los esfuerzos realizados para el diagnóstico de patologías mediante dichas señales.

An EMG-based feature extraction method using a normalized weight vertical visibility algorithm for myopathy and neuropathy detection (Patcharin, Sivarit, & Kosin, 2016)

Utilizan clasificadores potentes, como el vecino k-más cercano, la red neuronal de perceptrón multicapa y los clasificadores de máquinas de vectores de apoyo, para diferenciar señales de casos de salud, miopatía y ELA. Se llevan a cabo experimentos de evaluación del desempeño y los resultados revelaron una precisión del 98,36%, lo que corresponde a una mejora de aproximadamente un 2% en comparación con los métodos convencionales. Este artículo tuvo resultados excelentes.

Automatic diagnosis of neuromuscular diseases from electromyographic (EMG) records
(Sadi-Ahmed, Messekher, Namane, Kedir-Talha, & Mekaoui, 2017)

Hace un diagnóstico de enfermedades Neuromusculares, realizada de manera automática sin asistencia y sin segmentación. El método utilizado se basa en la transformada de Huang-Hilbert (HHT), que es un algoritmo basado en datos que descompone señal de forma natural sin conocimiento previo. Se obtuvo una tasa de precisión del 96,08% para la clasificación de señales EMG patológicas. Sin embargo, miopáticos y las patologías neuropáticas fueron discriminadas con una media tasa de precisión de clasificación del 100%. Estos resultados demostraron la eficiencia de su selección de características procedimiento y su idoneidad para reconocer el origen de la patología automáticamente. Se utilizaron 117 señales EMG para las pruebas.

Classification of EMG signals for assessment of neuromuscular disorder using empirical mode decomposition and logistic regression (Khan M. , Aziz, Bilal, & Aamir, 2019)

Diseño de un sistema de diagnóstico preciso y computacionalmente eficiente sigue siendo un desafío debido a la variación de las señales EMG tomadas de diferentes músculos con diferentes niveles de inserción de la aguja. Este estudio ofrece un marco completo para la clasificación precisa de señales EMG que incluye la eliminación de ruido por descomposición en modo empírico (EMD), extracción de características de los dominios de tiempo y frecuencia y clasificación por regresión logística (LR) y máquina de vectores de soporte (SVM). Este trabajo discrimina eficientemente entre la señal EMG de un sujeto sano y un paciente con enfermedad de ELA, independientemente del músculo que se utilice para la adquisición de la señal EMG y del nivel de inserción de la aguja. Las medidas de evaluación del desempeño como la sensibilidad, la especificidad, la medida F, la precisión de la clasificación total y el área bajo la curva ROC (AUC) se utilizan para validar el desempeño de ambos clasificadores. La técnica de clasificación LR muestra un rendimiento superlativo con una precisión de clasificación del 95,1%.

1.3.1 Posicionamiento de electrodos superficiales

De acuerdo con la SENIAM para las recomendaciones de posicionamiento de los electrodos, se muestran dos imágenes con 8 posiciones diferentes para el muestreo de las señales EMG.

Posicionamiento de electrodos vista frontal

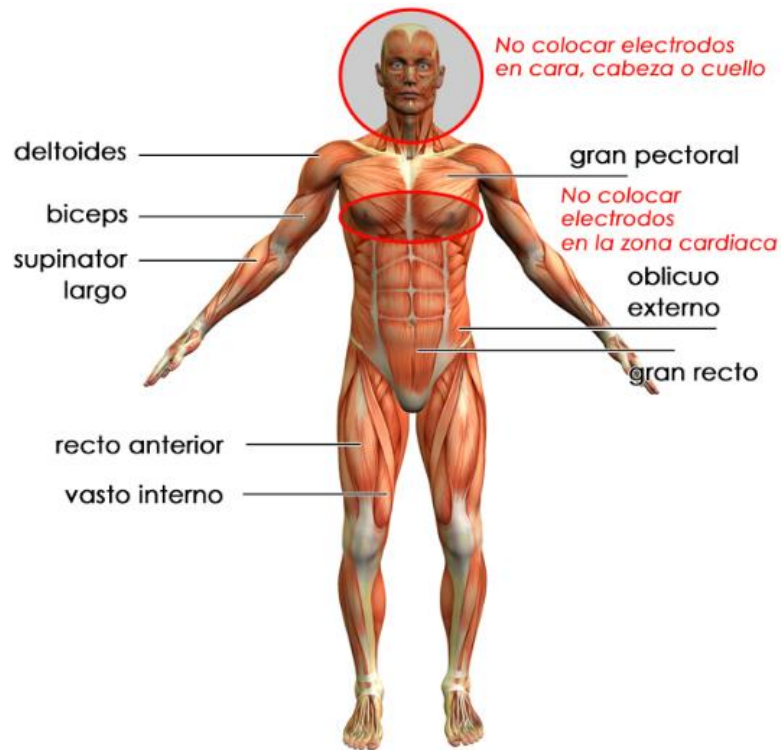


Figura 8. Vista frontal para posicionamiento de electrodos de aguja y superficie (Institut, 2011)

La Figura 8 y Figura 9 muestran los músculos relacionados con enfermedades neuromusculares, así como las ubicaciones que se han tomado en cuenta para este trabajo de tesis.

Vista Dorsal

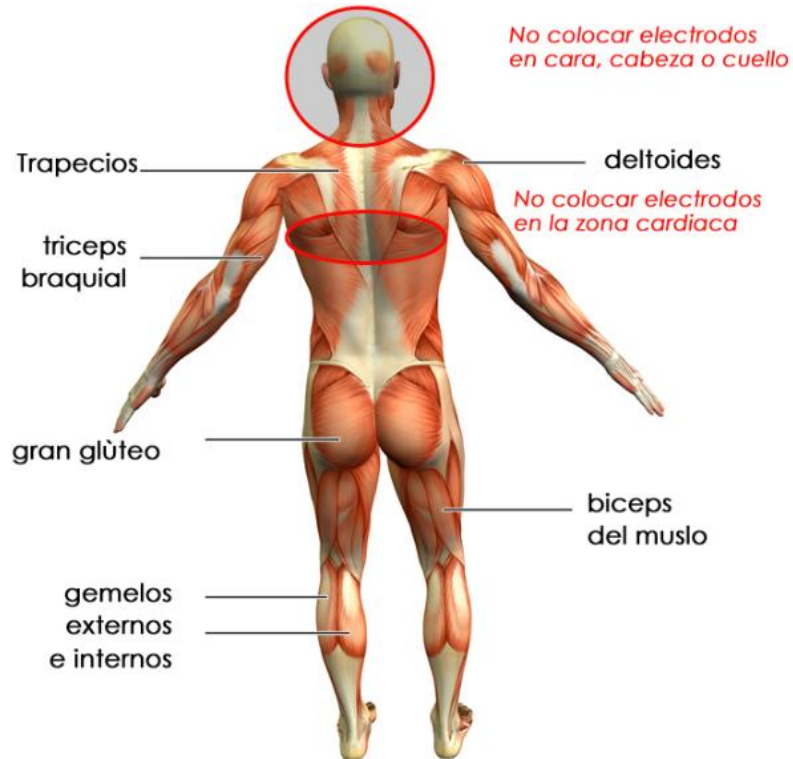


Figura 9. Vista dorsal del posicionamiento anatómico de los sitios seleccionados tanto de electrodo de aguja como de electrodo de superficie (Institut, 2011)

El posicionamiento de los sensores en los músculos frontales y dorsales de una persona está detallado en el Anexo 1 de este trabajo de tesis.

2. Metodología para Desarrollo del Sistema

Introducción

Las pruebas de electro diagnóstico son un importante método para distinguir entre desordenes musculares y de nervios. Una de las muchas técnicas es la electromiografía (EMG) y el Estudio de Conducción Nerviosa (del inglés nerve conduction studies NCS). La EMG y NCS proveen información en tiempo real acerca de lo que está ocurriendo fisiológicamente hablando con respecto al músculo y al nervio. Así como existen estudios de radiografía para ayudar a los médicos a diagnosticar, y que permite establecer un correcto diagnóstico en desordenes musculares. Una de las principales tendencias es el estudio de la EMG en diversos ámbitos. (González I. J., 2012)

2.1 Sistema neuromuscular

Los científicos han dividido el cuerpo humano en diferentes sistemas. Uno de ellos es el sistema neuromuscular, el cual está dividido en dos subsistemas: el sistema musculoesquelético y el sistema nervioso. (Subasi, Abdulhamit, 2019)

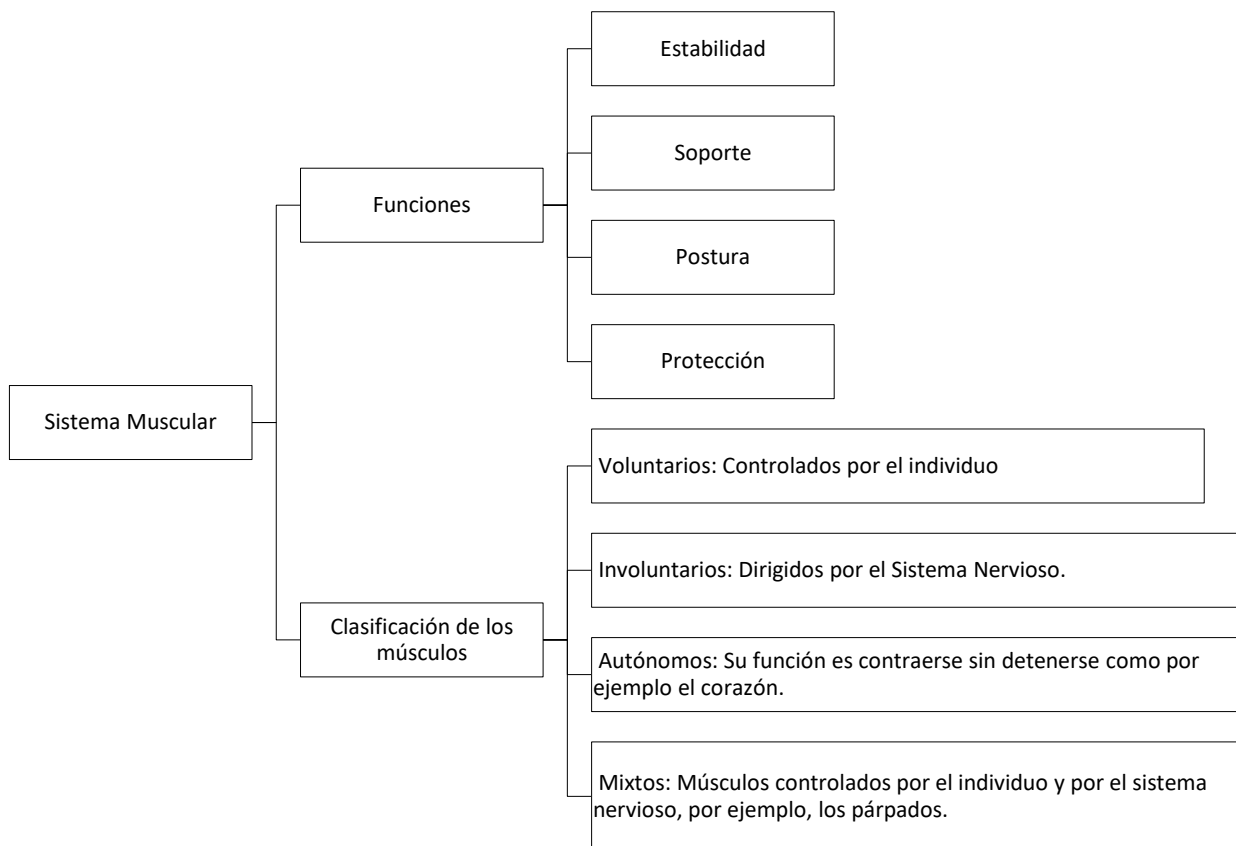


Figura 10. Clasificación del sistema muscular

La contracción humana es proveída por el sistema neuromuscular. El sistema nervioso central (SNC) controla los músculos a través de señales en los nervios. Los nervios actúan como un cable, llevando energía eléctrica desde el SNC a los músculos. Cuando las señales del SNC llegan al músculo, se produce un movimiento por contracción o relajación, que provoca el tirón o la liberación del hueso. EMG puede definirse como una técnica de trabajo, bajo condiciones neuromusculares basado en el potencial de acción celular mientras el músculo está activo. La amplitud (0.01-10mV en promedio) y la frecuencia (10-2000Hz en promedio) de las señales EMG depende de la localización, etiología, y tipo de patología.

Por ejemplo, aunque la duración del pulso de una señal EMG muestra la ubicación y la condición metabólica del músculo, los picos extraños pueden apuntar a la miopatía. El sistema neuromuscular también está bajo amenaza de enfermedad, como todos los demás sistemas del cuerpo humano. Estas enfermedades pueden originarse en el sistema nervioso, la unión neuromuscular y las fibras musculares.

Aunque algunas enfermedades neuromusculares causan debilidad en los músculos, algunas causan la muerte de neuronas o músculos. El diagnóstico preciso es muy importante para la recuperación exitosa del paciente que padece enfermedades neuromusculares. En muchos casos, las inspecciones clínicas no son suficientes para identificar la enfermedad y su ubicación. (Subasi, Abdulhamit, 2019)

Las señales EMG son interpretadas por un neurólogo que puede analizar las ondas usando conducción nerviosa, y es quien tiene conocimiento en las ondas de los músculos. En muchos casos, es crucial encontrar un experto que las interprete de manera correcta. Por lo tanto, diseñar un sistema de diagnóstico automatizado utilizando señales EMG se está volviendo importante cada día. Dicho requerimiento puede ser resuelto por técnicas de inteligencia artificial (IA).

Debido a lo anterior se implementó un sistema modular para ayudar a neurólogo a determinar desordenes neuromusculares basado en la clasificación de señales EMG usando técnicas de IA. Para diseñar un sistema automatizado de diagnóstico. Primero la señal EMG es pre procesada, y se extraen las características en el dominio del tiempo y frecuencia. En la etapa final de clasificación, los datos extraídos son usados como entradas para el clasificador basado en redes neuronales artificiales.

2.2 Diseño

El diseño general del sistema se muestra a continuación en la Figura 11, en donde se visualiza desde la adquisición de la señal EMG hasta el resultado final, que indica si la persona padece o no alguna enfermedad neuromuscular.

Los instrumentos para adquirir la señal EMG son electrodos simples, la señal es filtrada para reducir el ruido. Ya que la señal se digitaliza, ésta es guardada en la computadora para su posterior preprocesamiento. Ya que la señal ha sido filtrada, se hace unas operaciones matemáticas para extraer las características de la señal, cuando ya se tiene el módulo de la extracción de características completo, se procede a hacer la clasificación mediante las RNAs propuestas en el capítulo 3.

Adquisición de la señal EMG

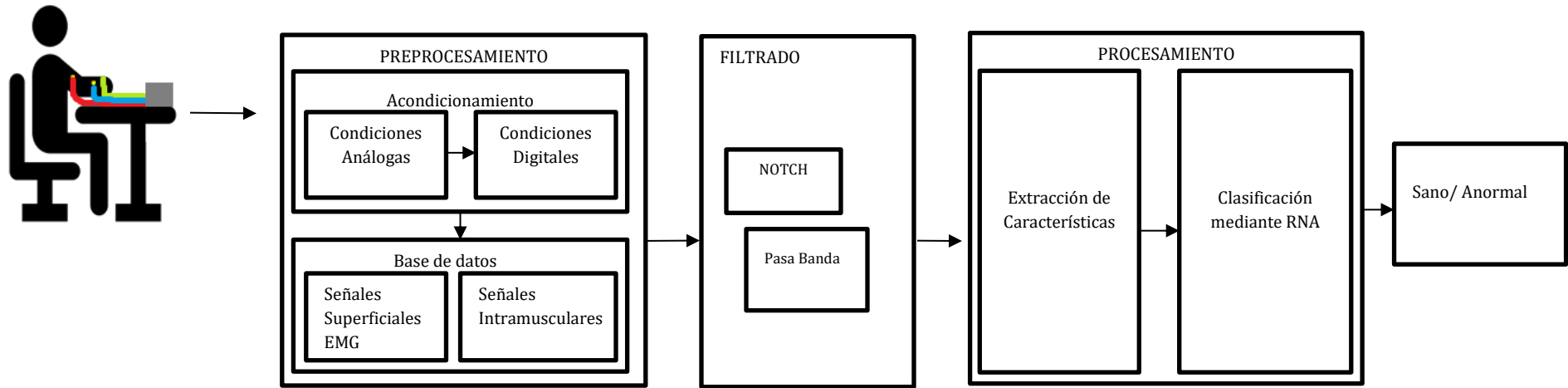


Figura 11. Diseño general de los módulos

2.3 Población

La base de datos contiene archivos binarios, .txt, .log, entre otros formatos. El tamaño de la muestra adquirida es de 74 pacientes entre sanos y con alguna neuropatía o miopatía. Los individuos muestreados se encuentran en un rango de edad de entre los 25 a 65 años incluidos hombres y mujeres. Las muestras se pueden encontrar en el anexo 6 de este documento de tesis.

2.4 Tipo de muestreo

El tipo de muestreo varía de acuerdo con las fuentes que se tomaron, a continuación, se muestra un ejemplo de la documentación de la base de datos en sus diversas fuentes.

2.5 Técnicas de procesamiento de señales

Existen diversas técnicas para el procesamiento de las señales, sin embargo, en este apartado se especifican solo las más utilizadas para diagnosticar miopatías y neuropatías. Estas técnicas están basadas en el uso de algoritmos matemáticos y métodos estadísticos, de manera que se pueda modificar la señal siendo más eficaces al momento de procesar la misma. Para ello, se determinan unos índices o parámetros cuantitativos concretos a partir del registro electromiográfico, que permiten realizar el diagnóstico en base a patrones normales conocidos. El primer paso es posicionar los electrodos donde se requiere hacer un muestreo de la señal, habiendo colocado los electrodos, se procede al paso dos, se registra la actividad electromiográfica y después se procede al paso tres, procesamiento de los datos analógicos para terminar con el procesamiento de la señal electromiográfica y la obtención de los rangos para su posterior evaluación. A continuación, se muestra el método que se usa actualmente para procesar la señal (M. Bodruzzaman, 1992) :



Figura 12. Método general para procesar la señal EMG

2.5.1 Registro de la actividad electromiográfica

El registro de la actividad eléctrica del cuerpo humano se da mediante la combinación de los potenciales de acción de las fibras musculares que componen una unidad motora (potenciales de acción de la unidad motora, PAUM) se puede detectar mediante electrodos superficiales localizados muy cerca del campo de estudio o mediante un electrodo de aguja insertado directamente en el músculo (inervación).

La activación muscular puede modelarse como un generador de impulsos que reflejan el SNC activando algunas unidades motoras, que a su vez actúan como filtros transformando cada impulso en una señal de PAUM. La señal EMG registrada será un tren de estas PAUMs. La Figura 13 muestra el proceso de adquisición de la señal EMG y su descomposición para obtener los PAUMs. (Raez, Mohd, & Faisal, 2006)

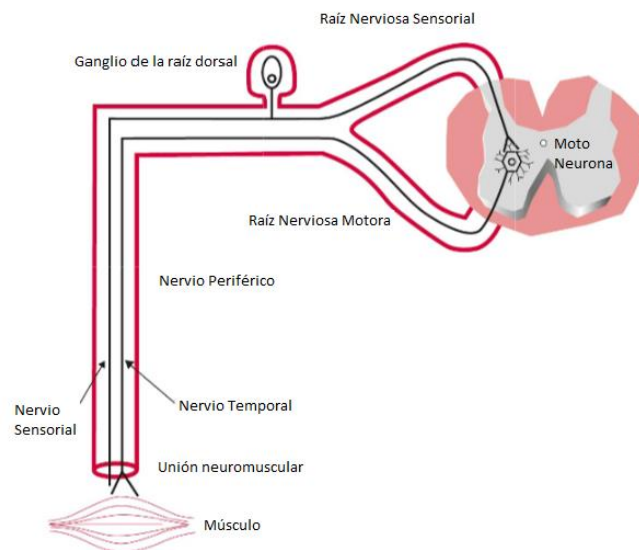


Figura 13. Fisiología de una motoneurona (Raez, Mohd, & Faisal, 2006)

El objetivo de los estudios electro diagnósticos es determinar si hay un problema a lo largo de la vía del sistema nervioso periférico y/o en los músculos y, de ser así, dónde está ubicado el problema.

La Figura 13, nos indica de acuerdo con las ubicaciones, posibles lesiones y diagnósticos asociados:

- Cuerpo de células nerviosas motoras (célula de la asta anterior) - esclerosis lateral amiotrófica
- Raíz - radiculopatía cervical o lumbar
- Axón - neuropatía tóxica
- Mielina - Síndrome de Guillain – Barré
- Unión neuromuscular - miastenia gravis
- Músculo - distrofia muscular

La exploración de la actividad EMG comprende el registro de la señal bioeléctrica en tres estados funcionales:

- En reposo, estando el músculo completamente relajado no debe registrarse ninguna actividad.
- Durante una contracción débil, activando un escaso número de Unidades Motoras se captan las descargas de los Potenciales de Acción de las Unidades Motoras correspondientes.
- Durante una contracción voluntaria máxima, que informa sobre la población de Unidades Motoras funcionando.

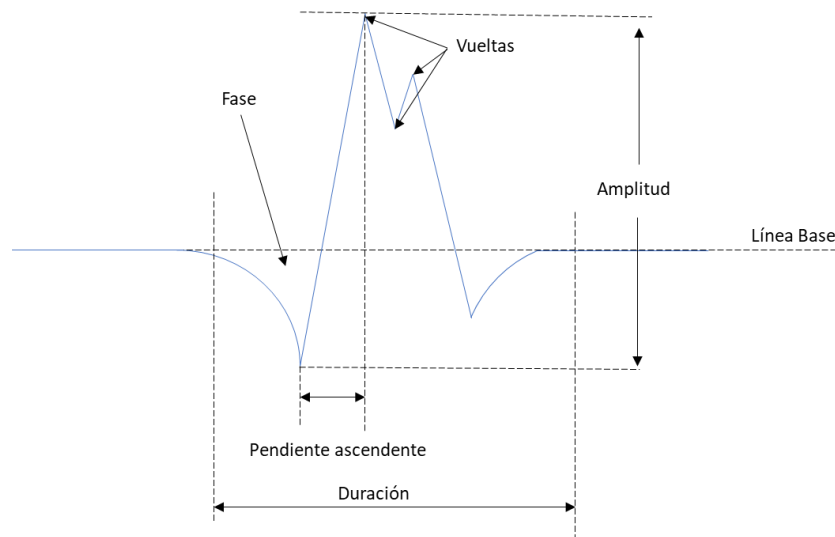


Figura 14. Potencial de la unidad motora (Cocero & Recuero, 1971)

La Figura 14 muestra la unidad motora con n fibras musculares. La suma de los potenciales de acción (PA) de cada fibra muscular ($PA_1 + PA_2 + \dots + PA_n$) captados por el electrodo concéntrico da lugar al potencial de unidad motora (PAUM). Se indican los principales parámetros que se miden sobre la forma de onda del PAUM: amplitud, duración, fase, turn o vueltas y línea de base. (Weiss, Weiss, & Silver, 2004)

2.5.2 Procesamiento de señales electromiográficas

Como resultado de la adquisición de las señales mioeléctricas, se visualizan los impulsos generados por las unidades motoras, el paso siguiente es realizar un análisis y procesamiento de la información generada.

Como es nuestro caso, la señal electromiográfica ha sido previamente pre procesada vía hardware, por lo tanto, los registros con los que se cuentan están documentados en el anexo 6.

2.5.2.1 Filtrado Digital

A pesar de haber hecho un acondicionamiento analógico de la señal con algunos componentes electrónicos y de haber limitado el ancho de banda de la señal, el ruido de los aparatos puede ser eliminados a través de un filtrado digital. Combinando filtros digitales, se pueden eliminar o limpiar las señales evitando alterar el contenido del resto de la señal EMG adquirida, además de que se pueden eliminar los ruidos que se pudieron pasar a través de los filtros convencionales que fallan a niveles bajos de amplitud de las EMG.

El filtro que se implementó para las señales EMG es:

- Pasa banda: deja pasar un determinado rango de frecuencias de una señal y atenúa el paso del resto.

Cabe destacar que, aunque este filtro fue implementado por hardware para la adquisición de las señales electromiográficas, se vuelve a filtrar digitalmente, sin embargo, al aplicarlo en Matlab la señal EMG se mejora ligeramente.

2.5.2.2 Rectificado y promedio de la señal

Este es un proceso en el que se rectifica la onda convirtiendo las amplitudes negativas (-) a positivas (+). Con el fin de obtener los parámetros como la media, valor de pico, o área bajo la curva. Como se había visto en temas anteriores son básicos para mostrar la actividad de las unidades motoras (UM) durante un periodo de tiempo determinado.

2.6 Extracción de características

La información se evalúa rápidamente y es extraída de cada segmento de tiempo para crear un conjunto de características que representan el patrón de la señal EMG. Entre las características más importantes son del punto 2.5.2.3.1 al 2.5.2.3.19. Estas características se basan en la amplitud de la señal, que puede considerarse como la desviación estándar variable en el tiempo de la señal, proporcional al número de MUAPs activas y su velocidad de disparo. Las características calculadas proporcionan información sobre la energía de forma de onda, amplitud, duración y frecuencia.

La extracción de características en señales EMG puede ser computada en 3 dominios: 1) Dominio del tiempo, 2) Dominio de la frecuencia y 3) Frecuencia-Tiempo o dominio de escala del tiempo (Jarušek & Kocian, 2018):

2.6.1 Valor absoluto medio mejorado (EMAV)

Estima el valor absoluto medio mejorado de la señal EMG. Como resultado en los cambios de la señal indica fatiga del músculo. Esta característica se encuentra en el dominio del tiempo (Jingwei, Abdul, Norhashimah, & Weihown, 2019). Está dada por:

$$\mathbf{EMAV} = \frac{1}{L} \sum_{i=1}^L |(x_i)^p|$$

$$p = \begin{cases} 0.75, & \text{si } i \geq 0.2L \text{ and } i \leq 0.8L \\ 0.5, & \text{otherwise} \end{cases}$$

(1)

Dónde:

X_i : *Coficiente wavelet*

L : *longitud de coeficientes*

p : *se usa para identificar la influencia de la muestra dentro de la señal.*

2.6.2 Longitud de onda mejorada (EWL)

Es una versión modificada de la longitud de onda y el valor absoluto medio.

$$EWL = \sum_{i=2}^L |(x_i - x_{i-1})^p|$$

$$p = \begin{cases} 0.75, & \text{si } i \geq 0.2L \text{ and } i \leq 0.8L \\ 0.5, & \text{otherwise} \end{cases} \quad (2)$$

Dónde:

X_i : *Coficiente wavelet*

L : *longitud de coeficientes*

p : *se usa para identificar la influencia de la muestra dentro de la señal.*

2.6.3 Valor absoluto medio (MAV)

Es un promedio del valor absoluto de la amplitud de la señal EMG en un segmento (Jingwei, Abdul, Norhashimah, & Weihown, 2019). Está dada por:

$$MAV = \frac{1}{L} \sum_{n=1}^L |Z_n|$$
(3)

Dónde:

Z_n : *valor de cada parte del segmento (Coficiente wavelet)*

L : *longitud de coeficientes*

2.6.4 Longitud de forma de onda (WL)

Es una frecuencia usada frecuentemente como característica de la señal EMG, representa la longitud acumulativa de la forma de paso en el tiempo (Jingwei, Abdul, Norhashimah, & Weihown, 2019). Está dada por:

$$WL = \sum_{n=1}^L |Z_n - Z_{n-1}|$$

(4)

Dónde

Z_n : valor de cada parte del segmento (Coeficiente wavelet)

L : longitud de coeficientes

2.6.5 Cruce cero (ZC)

Es una cantidad de veces que la forma de onda EMG cruza los ejes de amplitud cero (Boostani & Moradi, 2003) (M. Zardoshti & Afshordi, 1995) (Jingwei, Abdul, Norhashimah, & Weihown, 2019). Está dada por:

$$ZC = \sum_{i=1}^{L-1} f(x_i)$$

$$f(x_i) = \begin{cases} 1, & \text{si } \{(x_i > 0 \text{ and } x_{i+1} < 0) | (x_i < 0 \text{ and } x_{i+1} > 0)\} \\ & \text{and } |x_i - x_{i+1}| \geq T \\ 0, & \text{otherwise} \end{cases}$$

(5)

Dónde

x : Coeficiente wavelet

L : longitud de coeficientes

T = umbral debe incluirse en el cálculo de ZC para reducir el efecto del ruido (Hudgins, 1993)

2.6.6 Cambio de señal de pendiente (SSC)

También llamado el número de turnos (NT) es una cantidad de veces que la pendiente de la forma de onda EMG cambia de signo. Este método utiliza tres amplitudes de señal EMG consecutivas para detectar la pendiente de la forma de onda EMG (Jingwei, Abdul, Norhashimah, & Weihown, 2019). Está dada por:

$$SSC = \sum_{i=2}^{L-1} f(x_i)$$

$$f(x_i) = \begin{cases} 1, & \text{si } \{(x_i > 0 \text{ and } x_{i+1} < 0) | (x_i < 0 \text{ and } x_{i+1} > 0)\} \\ & \text{and } |x_i - x_{i+1}| \geq T \\ 0, & \text{otherwise} \end{cases}$$

(6)

Dónde

x: Coeficiente wavelet

L: longitud de coeficientes

T = umbral debe incluirse en el cálculo de SSC para reducir el efecto del ruido (Hudgins, 1993)

2.6.7 Media cuadrática (RMS)

Es una raíz cuadrada del promedio del cuadrado de los valores de amplitud de la señal EMG. En otras palabras, tiene un valor similar a la desviación estándar de la amplitud de la señal EMG. Es una característica que describe la fuerza del músculo y la contracción sin fatiga (Jingwei, Abdul, Norhashimah, & Weihown, 2019). Está dado por:

$$RMS = \sqrt{\frac{1}{L} \sum_{n=1}^L Z_i^2}$$

(7)

Dónde

Z_i : valor de cada parte del segmento (Coeficiente wavelet)

L : longitud de coeficientes

2.6.8 Cambio de amplitud promedio (DAMV)

También llamada difference absolute mean value (DAMV) es un promedio del valor absoluto de la diferencia entre las amplitudes EMG adyacentes en el segmento de tiempo (Ankoom, Quaine, & Laurillan, 2012) MAV de la longitud de onda EMG y, a veces, se llama "AAC de cambio de amplitud promedio" (Jingwei, Abdul, Norhashimah, & Weihown, 2019). Está dado por:

$$DAMV = \frac{1}{L} \sum_{i=1}^{L-1} |x_{i+1} - x_i| \quad (8)$$

Dónde

x : Coeficiente wavelet

L : longitud de coeficientes

2.6.9 Diferencia valor de desviación estándar absoluto (DASDV)

Es una raíz cuadrada del promedio del cuadrado de diferencia entre las amplitudes EMG adyacentes en el segmento de tiempo (Jingwei, Abdul, Norhashimah, & Weihown, 2019).

Es el RMS de la longitud de onda EMG. Está dado por:

$$DASDV = \sqrt{\frac{\sum_{i=1}^{L-1} (x_{i+1} - x_i)^2}{L - 1}} \quad (9)$$

Dónde

x : Coeficiente wavelet

L: longitud de coeficientes

2.6.10 Detector de registro (LD)

Es una característica que es buena estimando la fuerza ejercida (Jingwei, Abdul, Norhashimah, & Weihown, 2019). Está dado por:

$$LD = \exp\left(\frac{1}{L} \sum_{i=1}^L \log(|x_i|)\right)$$

(10)

Dónde

x: Coeficiente wavelet

L: longitud de coeficientes

2.6.11 Valor absoluto medio modificado 1(MMAV1)

Es una extensión de la característica MAV asignado la función de la ventana de peso (Jingwei, Abdul, Norhashimah, & Weihown, 2019). Está dado por:

$$MMAV1 = \frac{1}{L} \sum_{i=1}^L w_i |x_i|$$

(11)

$$w_i = \begin{cases} 1, & \text{si } 0.25L \leq i \leq 0.75L \\ 0.5, & \text{otherwise} \end{cases}$$

Dónde

x: Coeficiente wavelet

L: longitud de coeficientes

2.6.12 Valor absoluto medio modificado 2 (MMAV2)

Es otra extensión de la característica MAV asignando pesos continuos a la ventana de función (Jingwei, Abdul, Norhashimah, & Weihown, 2019). Está dado por:

$$MMAV2 = \frac{1}{N} \sum_{i=1}^N w_i |x_i|$$

(12)

$$w_i = \begin{cases} 1, & \text{si } x \leq i \leq thr \\ \frac{4i}{N}, & \text{si } i < 0.25N \\ \frac{4(i-N)}{N}, & \text{otherwise} \end{cases}$$

Dónde

x: Coeficiente wavelet

N: longitud de coeficientes

2.6.13 Porcentaje de miopulso (MYOP)

Es definido como el promedio de la señal del miopulso en el cual el valor absoluto de la señal EMG excede el valor del problema predefinido (Jingwei, Abdul, Norhashimah, & Weihown, 2019). Está dado por:

$$MYOP = \frac{1}{N} \sum_{i=1}^N f(x_i)$$

(13)

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

Dónde

x es el coeficiente wavelet

T es el valor umbral

N es la longitud del coeficiente.

2.6.14 Integral cuadrada simple (SSI)

Se define como la suma de los valores cuadrados de la amplitud de la señal EMG (Jingwei, Abdul, Norhashimah, & Weihown, 2019). Está dado por:

$$SSI = \sum_{i=1}^N (|x_i^2|)$$

(14)

Dónde

x: Coeficiente wavelet

N: longitud de coeficientes

2.6.15 Varianza (VAR)

La varianza de EMG (VAR) es un promedio del valor cuadrado de la amplitud de la señal EMG (Jingwei, Abdul, Norhashimah, & Weihown, 2019). Está dado por:

$$VAR = \frac{1}{N-1} \sum_{i=1}^N x_i^2$$

(15)

Dónde

x: Coeficiente wavelet

N: longitud de coeficientes

2.6.16 Amplitud willinson (WA)

Actúa como un indicador del disparo de los potenciales de la unidad motora (Jingwei, Abdul, Norhashimah, & Weihown, 2019). Está dado por:

$$WA = \sum_{i=1}^{L-1} f(x_i) \quad (16)$$

$$f(x_i) = \begin{cases} 1, & \text{si } |x_i - x_{i+1}| \geq T \\ 0, & \text{otherwise} \end{cases}$$

Dónde

x: Coeficiente wavelet

L: longitud de coeficientes

2.6.17 Longitud máxima del fractal (MFL)

Se propone como una característica, que puede medir patrones de señal EMG en contracción muscular de bajo nivel (Arjunan, 2008). Se define como la longitud fractal promedio de la señal medida. Mide la activación a bajo nivel de la contracción del músculo (Jingwei, Abdul, Norhashimah, & Weihown, 2019). Está dada por:

$$MFL = \log_{10} \left(\sum_{n=1}^{L-1} |Z_n - Z_{n+1}| \right)$$

Dónde

Z_n : valor de cada parte del segmento (Coeficiente wavelet)

L: longitud de coeficientes

(17)

2.6.18 Frecuencia mediana (MDF)

Es una frecuencia a la que el espectro de potencia EMG se divide en dos áreas con una potencia total igual (Phinyomark, Thongpanja, Hu, Phukpattaranont, & Limsakul, 2013).

Está dada por:

$$\sum_{j=1}^{MDF} P_j = \sum_{j=MDF}^M P_j = \frac{1}{2} \sum_{j=1}^M P_j$$

Dónde

x: Coeficiente wavelet

L: longitud de coeficientes

2.6.19 Frecuencia media (MNF)

Es una frecuencia promedio, que se calcula como la suma del producto del espectro de potencia EMG y la frecuencia dividida por la suma total del espectro de potencia EMG. Hay otras formas de llamar a esta característica, como la frecuencia central y el centro de gravedad espectral (Phinyomark, Thongpanja, Hu, Phukpattaranont, & Limsakul, 2013). Está dada por:

$$MNF = \frac{\sum_{i=1}^M f_i PSD_i}{\sum_{i=1}^M PSD_i}$$

(19)

$$f_i = \frac{i - \text{sampling rate}}{2M}$$

Dónde

f_i = es la frecuencia del espectro en una frecuencia

bin i, PSD_i = es el espectro de potencia EMG en una frecuencia

bin j, y M = es la longitud de la frecuencia completa bin.

2.6.20 Duración

Para encontrar la duración de los MUAP, se toma como duración el inicio y el final del MUAP identificado deslizando una ventana de medición de 1 ms de longitud y 10 microvoltios de ancho.

$$\text{Duración} = \frac{i1 - i2}{\text{frecuencia de muestreo}}$$

(20)

Dónde:

i1, i2 son puntos de inicio y final de la muestra de la MUAP

2.6.21 Duración del pico (DP)

La duración del pico es el periodo desde el primer pico positivo hasta el último pico positivo del MUAP (Goen, 2014).

$$\text{DP} = \frac{k_1 - k_2}{\text{frecuencia de muestreo}}$$

(21)

Dónde

k_i = es la frecuencia del espectro en una frecuencia

Se toma el punto k^{th} y se compara el punto de procedimiento $(k - 1)$ y el punto siguiente $(k + 1)$. Si la magnitud absoluta del punto k^{th} es mayor que $k - 1^{th}$ y $k + 1^{th}$, después se almacena el punto k^{th} como un pico. Deja que k_1 y k_2 sean el primer y último punto + *ve* picos identificados.

2.6.22 Amplitud

La diferencia entre el máximo pico positivo y el pico mínimo negativo es tomada como amplitud de la MUAP (Goen, 2014).

$$\text{Amplitud} : M1 + M2$$

(22)

Dónde

M_i = Magnitud

2.6.23 Área

Es la suma del MUAP rectificado integrado en la medida de duración (Goen, 2014).

De acuerdo con lo anterior, las características del punto 2.6.1 a la 2.6.19 y los rangos sugeridos deben ser considerados para la adquisición de la señal EMG: (Durán Acevedo , 2014)

- Espectro de la señal: 0 - 500 Hz
- Energía dominante: 50 –150 Hz
- Ruido de línea (60Hz).
- Señal aleatoria: Representación por medio de distribución gaussiana.
- Amplitudes: 0 – 6 mV pico y 0 - 1.5 mV (RMS)

El tiempo constante debe ser consistente con la naturaleza de la actividad que está siendo realizada.

- Movimiento lento: Usar tiempo mayor.
- Movimiento rápido: Usar menor tiempo.

Después de haber extraído las características de las señales, se hace una selección de datos como entrada, los targets y el set de entrenamiento mediante RNA como método de clasificación.

2.7 Clasificación mediante redes neuronales artificiales

Se considera como inteligencia artificial como un conjunto de técnicas, algoritmos y herramientas que nos permiten resolver problemas para los que, a priori, es necesario cierto grado de inteligencia, en el sentido de que son problemas que suponen un desafío incluso para el cerebro humano. (García, 2018)

Una red neuronal artificial (RNA) es un modelo conexionista cuyos elementos o nodos, conectados entre sí, simulan las funciones desempeñadas por las células cerebrales denominadas neuronas.

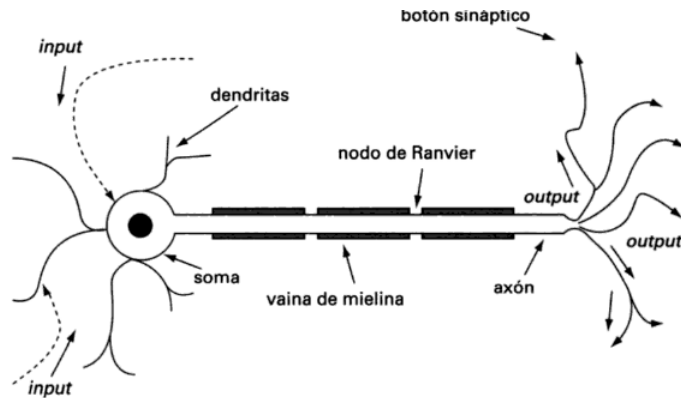


Figura 15. Características de la neurona (Foundation, 2020)

Las RNAs están formadas por una serie de procesadores elementales, denominadas neuronas artificiales, que constituyen dispositivos simples de cálculo que, a partir de un vector de entrada procedente del mundo exterior, bien a partir de estos estímulos recibidos de otras neuronas, proporcionan una respuesta única (salida).

Existen 3 tipos de neuronas artificiales:

2.7.1 Neuronas de entrada

Reciben señales desde el entorno, provenientes de sensores o de otros sectores del sistema (como archivos de almacenamiento de patrones de aprendizaje).

2.7.2 Neuronas de salida

Envían su señal directamente fuera del sistema una vez finalizado el tratamiento de la información (salidas en la red).

2.7.3 Neuronas ocultas

Reciben estímulos y emiten salidas dentro del sistema, sin mantener contacto alguno con el exterior. En ellas se lleva a cabo el procesamiento básico de la información, estableciendo la representación interna de ésta.

Como se muestra en la Figura 16, las tres capas de la red neuronal descritas previamente en los puntos 2.6.1, 2.6.2 y 2.6.3 nos muestran las letras x como datos de entrada, las letras W como los pesos, y las letras Y simbólicamente como salidas.

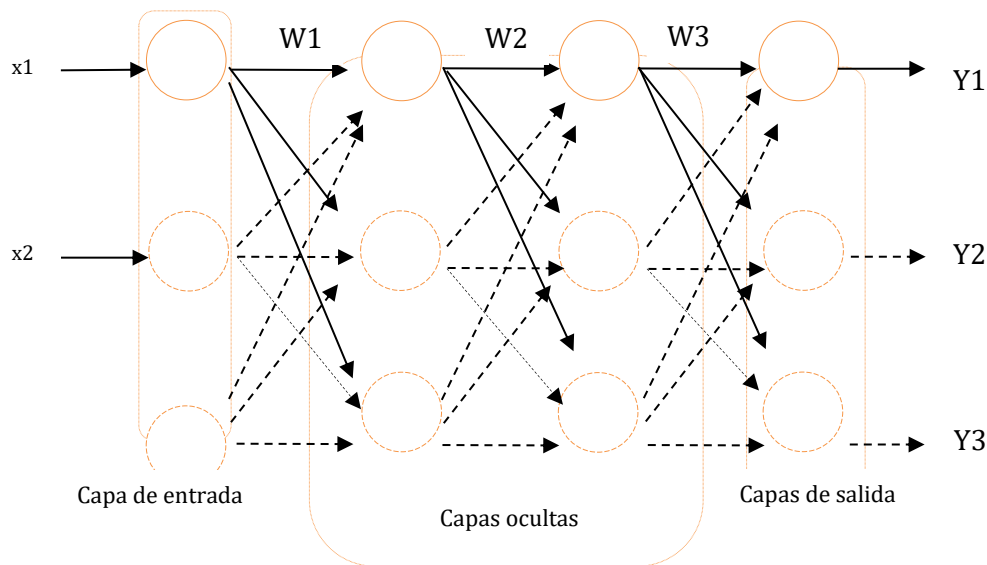


Figura 16. Definición de la red neuronal

Actualmente para lograr una mejor aproximación, se cuenta con una red neuronal artificial para clasificar las señales EMG en normal, y miopático/neuropático y lograr el objetivo del proyecto de tesis. En la Figura 17 se describe la metodología adaptada para resolver el problema. Se ingresan los datos al sistema, el cual hace la extracción de características de las señales EMG, de los cuales, se usan como entradas para el clasificador, previamente se le indican los resultados esperados (señal sana, señal con neuropatía/miopatía) y arroja los resultados.

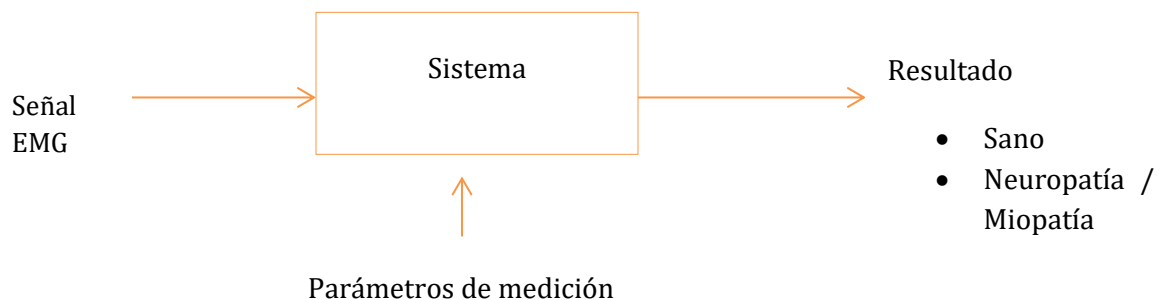


Figura 17. Módulo general metodología de diagnóstico

Existen diferentes métodos para el diagnóstico de enfermedades neuromusculares, a lo largo del desarrollo de este proyecto de tesis se probaron diferentes técnicas y métodos para obtener el diagnóstico de las enfermedades neuromusculares alimentada con diversas

señales EMG que prueban la capacidad de la red para tomar decisiones y clasificar de manera adecuada las señales, de tal forma que sea capaz de decidir si alguna de las mismas tiene patrones de enfermedades neuromusculares o son sanas.

2.8 Características de los patrones patológicos

Características de los patrones patológicos de la actividad espontánea en señales EMG y las posibles enfermedades asociadas.

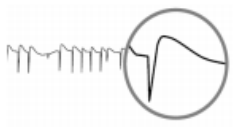

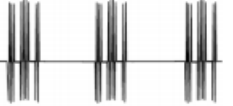
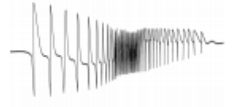
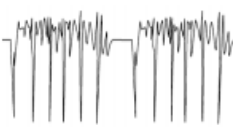
Patrón patológico en la señal EMG	Características de la señal	Miopatías asociadas	Neuropatías asociadas
 ondas agudas positivas	Frecuencia: 0.5-15Hz Amplitud: 20-1000 μ V Duración: 10-30 ms	<ul style="list-style-type: none"> - Miopatías Inflamatorias - Distrofias musculares - Miositis por cuerpos de inclusión - Miopatías congénitas - Rabdomiólisis - Trauma muscular - Triquinosis 	<ul style="list-style-type: none"> - Radiculopatía - Neuropatía periférica axonal - Plexopatías - Neuropatías por atrapamiento - Enfermedad de la motoneurona - Mononeuropatías
 Fibrilaciones	Frecuencia: 0.5-15Hz Amplitud: 20-1000 μ V		
 Descarga mioquímica	Forma continua: Frecuencia: 5-10Hz Forma Discontinua: Frecuencia: 0.1-10Hz	<ul style="list-style-type: none"> - Parálisis de Bell - Esclerosis Múltiple - Polirradiculopatía 	<ul style="list-style-type: none"> - Cuadro de radiación plexopática. - Lesiones nerviosas crónicas
 Descarga Miotónica	Frecuencia: 20-100Hz	<ul style="list-style-type: none"> - Distrofia miotónica - Miotomía compleja - Paramiotonía - Parálisis periódica hiperpotasemia - Polimiositis - Deficiencia de maltasa ácida 	<ul style="list-style-type: none"> - Radiculopatía crónica. - Polineuropatía axonal crónica.
 Descargas repetitivas complejas	Frecuencia: 10-100Hz Amplitud: 50-500 μ V De inicio y pausa rápida.	<ul style="list-style-type: none"> - Miopatías inflamatorias - La distrofia muscular del anillo óseo. - Mixedema - Síndrome de Schwartz-Jampel 	<ul style="list-style-type: none"> - Neuropatías crónicas o radiculopatías - Poliomielititis - Atrofia muscular espinal - Enfermedad de la motoneurona - Neuropatías hereditarias

Tabla 4. Características de la señal miopática y neuropática. (Osuna , González, & Ilzarbe, 2012)

Sin embargo y a pesar de los análisis presentados el sistema solo predice si una persona está enferma o sana como se observa en la Figura 17. Como trabajo futuro se propone la predicción de las enfermedades a detalle.

2.9 Diagnóstico de trastornos neuromusculares con señales EMG

Existen dos grandes tipos de patrones electromiográficos: los patrones neurógenos; comunes para afecciones propias de las motoneuronas, raíces nerviosas, plexos nerviosos y nervios periféricos, y los patrones miógenos: los patrones generados por afecciones musculares, como se muestra en la Tabla 4. En el patrón neurógeno, la destrucción de una motoneurona se traduce a nivel muscular en la pérdida de inervación por parte de una unidad motora. Las unidades motoras remanentes tienen que descargar a una frecuencia mayor para generar niveles adecuados de fuerza muscular. El patrón neurógeno posee tres características básicas: *empobrecimiento del trazado a la contracción máxima, el aumento del reclutamiento espacial y el aumento de amplitud y duración de los potenciales de unidad motora*. En el patrón miógeno, sin embargo, la lesión primaria está ubicada en la fibra muscular que degenera y se sustituye por tejido fibroso. No hay lesión ni destrucción de axones. El resultado es la visualización de unos pocos potenciales, ya que menos fibras contribuirán a la unidad motora afectada. La pérdida de fibras musculares provoca descarga de múltiples unidades motoras para genera una pequeña cantidad de fuerza muscular. Los potenciales miopáticos aparecen por tanto reducidos en amplitud y en duración.

3. Técnicas de Reconocimiento de Patrones

3.1 Introducción

¿Qué es el reconocimiento de patrones? Originalmente proviene de la neurociencia cognitiva y la psicología, es un proceso cognitivo que ocurre en nuestro cerebro cuando unimos cierta información que encontramos con los datos almacenados en nuestra memoria cerebral.

Tanto en las Ciencias Computacionales y Machine Learning (Aprendizaje automático) el reconocimiento de patrones es una tecnología que hace coincidir la información almacenada en la base de datos con los datos entrantes.

El reconocimiento de patrones es la ciencia de hacer inferencias a partir de datos perceptuales, utilizando herramientas de estadísticas, probabilidad, geometría computacional, aprendizaje automático, procesamiento de señales y diseño de algoritmos. Por lo tanto, es de importancia central para la inteligencia artificial y la visión por computadora, y tiene aplicaciones de largo alcance en ingeniería, ciencia, medicina y negocios. (Tohka & Ruotsalainen, 2012)

Dos enfoques del reconocimiento de patrones

Reconocimiento de Patrones	Aprendizaje Automático
El reconocimiento de patrones es la aplicación de ingeniería de varios algoritmos con el propósito de reconocer patrones en los datos.	Es un método de análisis de datos que automatiza la construcción de modelos analíticos.
El reconocimiento de patrones es más teórico	El aprendizaje automático es más práctico
Puede ser un problema en tiempo real	Puede ser una solución para un problema en tiempo real
El reconocimiento de patrones puede estar fuera de la máquina.	Necesitamos máquinas / computadoras para aplicar algoritmos de Aprendizaje Automático

Tabla 5. Enfoque de las técnicas de reconocimiento de patrones y de aprendizaje automático

En la Tabla 5 se visualizan los dos enfoques del problema que se tiene y dos posibles acercamientos al resultado.

3.2 Reconocimiento de patrones

Existen diferentes enfoques para el reconocimiento de patrones:

- Reconocimiento estadístico de patrones. Este enfoque se basa en la teoría de probabilidad y estadística y supone que se tiene un conjunto de medidas numéricas con distribuciones de probabilidad conocidas y a partir de ellas se hace el reconocimiento.
- Reconocimiento sintáctico de patrones. Este enfoque se basa en encontrar las relaciones estructurales que guardan los objetos de estudio, utilizando la teoría de lenguajes formales. El objetivo es construir una gramática que describa la estructura del universo de objetos.
- Redes neuronales. Este enfoque supone que tiene una estructura de neuronas interconectadas que se estimulan unas a otras, las cuales pueden ser “entrenadas” para dar una cierta respuesta cuando se le presentan determinados valores.
- Reconocimiento lógico combinatorio de patrones. Este enfoque se basa en la idea de que la modelación del problema debe ser lo más cercana posible a la realidad de este, sin hacer suposiciones que no estén fundamentadas. Uno de los aspectos esenciales del enfoque es que las características utilizadas para describir a los objetos de estudio deben ser tratadas cuidadosamente.

El enfoque que se seleccionó para el diagnóstico de enfermedades neuromusculares mediante procesamiento de señales EMG es usando las RNAs pues este enfoque es ampliamente utilizado en aplicaciones que necesitan la clasificación o reconocimiento de patrones de señales EMG. (Tsuji, Fukuda, Kaneko, & Ito, 2000) (González, Soubervielle, Vital, & Pérez, 2012) (Bodruzzaman, Zein-Sabatto, Marpaka, & Kari, 1992) (Sadi-Ahmed, Messekher, Namane, Kedir-Talha, & Mekaoui, Automatic diagnosis of neuromuscular diseases from electromyographic (EMG) records, 2017) (Powar & Chemmangat, 2019) (Ahsan, Ibrahimy, & Khalifa, 2011) (Katsutoshi, Seiji, Koji, & Takao, 1993) (Elamvazuthi, y otros, 2015) (M. , G., & M., 2012) (Deepika & Wadhvani, 2012) (Khan M. U., Aziz, Bilal, & Aamir, 2019) (Karlik, 2014) (Patidar, Jain, & Parikh, 2013) (Laxmi & Sangeeta, 2012)

3.2.1 Componentes básicos de un sistema de reconocimiento de patrones

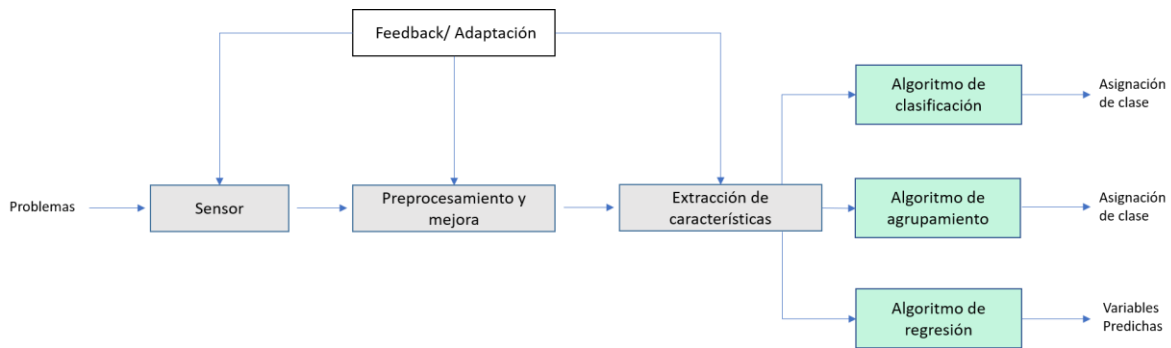


Figura 18. Enfoque sistemático al reconocimiento de patrones

El sistema de reconocimiento de patrones funciona para este problema puesto se cuentan con todos los elementos que lo contienen para el diagnóstico y clasificación de la señal EMG mediante un algoritmo de clasificación (como se detallan los módulos del sistema en la Figura 11). Se cuenta con la base de datos de las señales EMG extraídas, se pre procesa la señal EMG, se extraen las características, y con técnicas de la inteligencia artificial se clasifica en enfermo o sano.

3.2.2 Elementos del reconocimiento de patrones

El modelo clásico del reconocimiento de patrones incluye 3 operaciones mayores que son representación, extracción de características y clasificación, de manera simplificada. Este modelo permite la formulación y discusión de muchos problemas importantes y ayudó a formalizar el problema de la clasificación en la aplicación de este proyecto de tesis, el diagnóstico de enfermedades por medio de EMG. Para está aplicación se resolvieron dos problemas:

- El diseño del extractor de características
- Diseño del clasificador de patrones

3.2.2.1 Diseño del extractor de características

La función del extractor de características es extraer o medir las características importantes de los patrones de entrada. Las características extraídas se denominan *características* y determinan mejor todos los patrones de entrada posible. Para este caso, se cuentan con 19 características antes descritas en el punto 2.5.3 (Karyakarte & Savant, 2019).

3.2.2.2 Diseño del clasificador de patrones

Para el diseño del clasificador el sistema usa las características extraídas en la etapa anterior de cada uno de los patrones para reconocerlos y asociar cada uno a su clase. Hay dos tipos de clasificadores:

- Los clasificadores que contienen el conocimiento de cada categoría de patrón y también el criterio o métrica para discriminar entre clases de patrones, que pertenecen al aprendizaje supervisado.
- El aprendizaje no supervisado en el que los parámetros del sistema se adaptan utilizando solo la información de la entrada, y restringido por reglas internas preespecificadas, intenta encontrar patrones inherentes en los datos que luego pueden usarse para determinar el valor de salida correcto para nuevas instancias de datos.

El reconocimiento de patrones generalmente se clasifica de acuerdo con el tipo de aprendizaje utilizado para generar el valor de salida en este paso. Este paso nos permite reconocer un objeto o un patrón mediante el uso de algunas características derivadas de los pasos anteriores. Se asignó un valor de entrada para el vector de características en un conjunto de clases dado (en el caso del diagnóstico del sistema, para sano =1 y miopatía/neuropatía = 2). Existen diferentes métodos de clasificación utilizados mediante redes neuronales artificiales, los algoritmos que determinaron una eficacia mayor fueron feed-forward backpropagation y cascade-forward backpropagation, además se hicieron pruebas con los algoritmos probabilísticos y perceptrón.

3.3 Módulo clasificador propuesto mediante RNAs

Las redes neuronales artificiales (RNAs) es una rama destacada de la IA, surgen con la motivación de desarrollar algoritmos capaces de procesar información al igual que el cerebro humano, de ahí que en ocasiones se hable de la neurona biológica como inspiración de la neurona artificial. (Galván & Valls, 2018)

Como se habla en el punto 3.2, los investigadores han usado las redes neuronales artificiales para el procesamiento de bioseñales.

El trabajo se ha llevado a cabo con la red neuronal feed-forward backpropagation que contiene una capa oculta y un algoritmo de retro propagación (backpropagation) para el entrenamiento.

Las RNAs con diferentes números de neuronas ocultas fueron probadas (como se muestra en el tema 3.2.2) y validadas en la etapa del entrenamiento para probar la eficacia y predecir las clases de salida. Se simularon distintas redes con diferentes neuronas ocultas y ambos tipos de algoritmos de entrenamiento, las respuestas de la red y la función de las redes se verifican con análisis estadísticos de MSE.

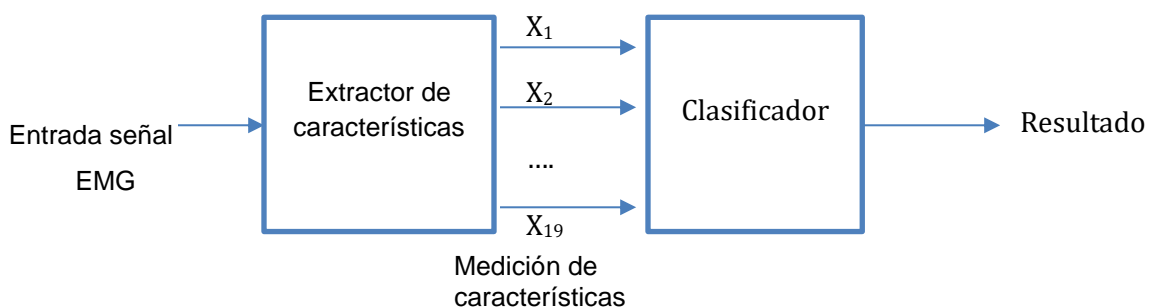


Figura 19. Sistema de reconocimiento de patrones

Muchos investigadores utilizan desde 3 hasta 12 características estadísticas de la señal, sin embargo, para mejorar la precisión hemos decidido utilizar 19, tras un análisis previo de la literatura para el diagnóstico de las enfermedades neuromusculares. (Tsuji, Fukuda, Kaneko, & Ito, 2000) (Sadi-Ahmed, Messekher, Namane, Kedir-Talha, & Mekaoui,

Automatic diagnosis of neuromuscular diseases from electromyographic (EMG) records, 2017) (Powar & Chemmangat, 2019) (Ahsan, Ibrahimy, & Khalifa, 2011) (Katsutoshi, Seiji, Koji, & Takao, 1993) (Elamvazuthi, y otros, 2015) (M. , G., & M., 2012) (Deepika & Wadhvani, 2012) (Khan M. U., Aziz, Bilal, & Aamir, 2019) (Karlik, 2014) (Patidar, Jain, & Parikh, 2013) (Laxmi & Sangeeta, 2012)

3.4 Aproximador backpropagation

Los diferentes tipos de estructuras de las RNAs han sido utilizados por muchos investigadores debido a su capacidad de adaptarse y aprender de varios datos arbitrarios para clasificar las señales EMG.

El perceptrón multicapa o backpropagation, es considerado un aproximador universal, es decir, cualquier función continua puede aproximarse mediante una RNA backpropagation, esto ha provocado que sea una de las arquitecturas más utilizadas en aplicaciones como, clasificación, diagnóstico médico, reconocimiento de caracteres, control de procesos, predicciones en series de tiempo, entre muchas otras aplicaciones; esto no quiere decir que este tipo de RNA sea la que ha dado mejores resultados ya que cada problema tiene características que lo definen y esto determinará el tipo de red a utilizar.

3.4.1 Arquitectura

La arquitectura de la red backpropagation consiste en una red multicapa, parcial o totalmente conectada con conexiones feed-forward y aprendizaje supervisado, normalmente es una arquitectura totalmente conectada, las modificaciones del problema original dependerán del problema a tratar.

Una red backpropagation se define formalmente:

Considere una red backpropagation con C capas, $C - 2$ capas ocultas y n_c neuronas en la capa c , para $c = 1, 2, 3, \dots, C$. Cada conexión tiene un peso el cual se denota con w , entonces $WC = (w_{ij}^c)$ será la matriz de los pesos asociados a las conexiones de la capa c con la capa $c + 1$ para $c = 1, 2, 3, \dots, C - 1$, donde w_{ij}^c representa el peso de la conexión de

la neurona i de la capa c con la neurona j de la capa $c + 1$ y sea $U^c = (u_i^c)$ el vector de umbrales de las neuronas de la capa c para $c = 2, \dots, C$. (González G. R., 2009)

La activación de la neurona i en la capa c se define como:

$$a_i^c = f\left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c\right) \quad i = 1, 2, 3, \dots, n_c; c = 2, 3, \dots, C - 1 \quad (23)$$

Del mismo modo las salidas de la red estarán definidas por:

$$y_i = a_i^c = f\left(\sum_{j=1}^{n_{c-1}} w_{ji}^{c-1} a_j^{c-1} + u_i^c\right) \quad i = 1, 2, \dots, n_c \quad (24)$$

Ambas ecuaciones utilizan la función f , a esta función se la llama función de activación, para backpropagation se pueden utilizar diversas funciones, pero principalmente destacan la sigmoideal y la tangente hiperbólica. Para el problema que se aborda, en el capítulo 4 se describe a detalle la función tangente sigmoideal.

Función	Expresión
Sigmoideal	$f(x) = \frac{1}{1 + e^{-x}}$
Tangente Hiperbólica	$f(x) = \frac{1 + e^{-x}}{1 + e^{-x}}$
Tangente Sigmoideal	$f(x) = \frac{2}{1 + e^{-2*x}} - 1$

Tabla 6. Funciones de activación comunes para la red backpropagation

El aprendizaje en la red backpropagation es un *aprendizaje supervisado*, en el cual se busca minimizar el error existente entre la salida de la red y la salida deseada, este error debe propagarse hacia las capas anteriores para modificar los pesos de las conexiones, por lo tanto, se debe distinguir que existen dos tipos de señales:

- Señales de función: inicialmente provenientes de la capa de entrada y propagándose de capa en capa hacia adelante hasta la capa de salida
- Señales de error: únicamente generadas en el proceso de aprendizaje para modificar los pesos, y propagándose hacia atrás

La definición formal para la señal de error para la neurona de salida j en la iteración n (n -ésimo patrón de entrenamiento) es:

$$e_j(\mathbf{n}) = \frac{1}{2}(d_j(\mathbf{n}) - y_j(\mathbf{n}))^2 \quad (25)$$

donde d_j es la salida deseada para la neurona de salida j y y_j es la salida de la red en la neurona de salida j . El error total producido por todas las neuronas de la capa de salida está definido por:

$$\xi(\mathbf{n}) = \frac{1}{2} \sum_{j \in C} e_j^2(\mathbf{n}) \quad (26)$$

donde el conjunto C contiene todas las neuronas de la capa de salida, y N como el número de patrones en el conjunto de entrenamiento, el error cuadrático promedio se representa:

$$\xi_{av} = \frac{1}{N} \sum_{n=1}^N \xi(\mathbf{n}) \quad (27)$$

El aprendizaje se debe realizar minimizando ξ_{av} , en la práctica y en este proyecto de tesis se utilizó el método de gradiente estocástico, donde se busca minimizar los errores de cada patrón $\xi(\mathbf{n})$ en lugar de minimizar ξ_{av} , la corrección de los pesos está definida por la regla Delta y se representa:

$$\Delta w_{ji}(\mathbf{n}) = -\eta \frac{\delta \xi(\mathbf{n})}{\delta w_{ji}} \quad (28)$$

Donde $\xi(\mathbf{n})$ se representa en la ecuación 10 y η es conocido como la razón o tasa de aprendizaje que influye en la magnitud del desplazamiento de la superficie de error. El signo negativo corresponde al gradiente descendiente, utilizando la ecuación 29 los pesos de las conexiones se modifican mediante la siguiente regla:

$$\mathbf{w}(n) = \mathbf{w}(n-1) - \Delta \mathbf{w} \quad (29)$$

Uno de los problemas que presenta este tipo de red, es el tiempo de entrenamiento provocando que los algoritmos utilizados para estimar el número de capas y neuronas ocultas tarden mucho en dar un resultado apropiado. Este paso es muy importante ya que el número adecuado de capas y neuronas ocultas aumenta la capacidad de la RNA para resolver un problema.

3.4.2 Arquitectura de la red neuronal artificial

Una RNA consiste en neuronas artificiales como se observa en la Figura 15, interconectadas y agrupadas por capas. Como se muestra en la Figura 20, la configuración de una red neuronal que se utiliza en este proyecto de tesis está representada por:

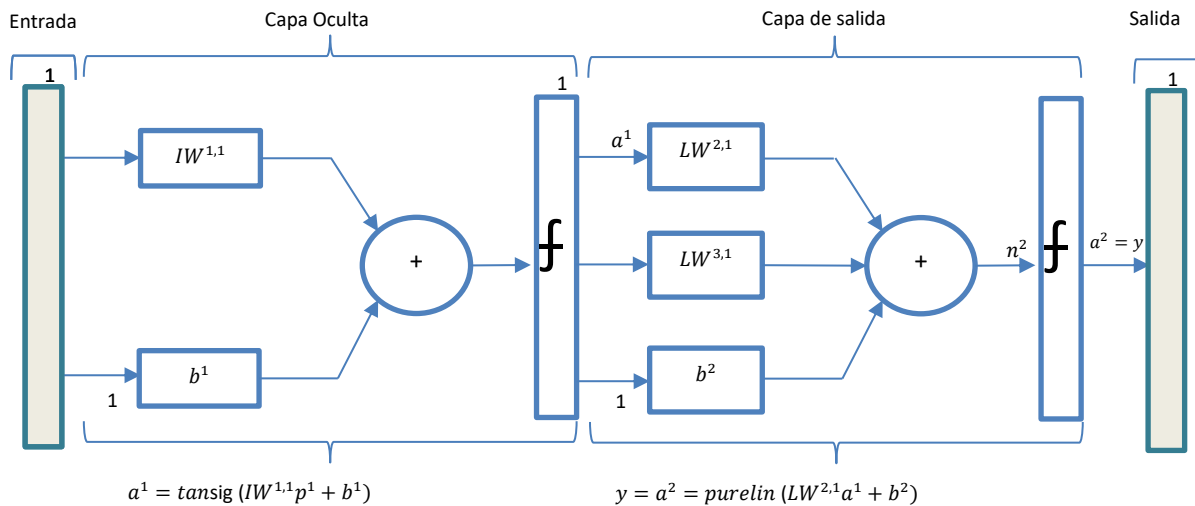


Figura 20. Arquitectura

La red consta de 3 capas:

- Entrada
- Capa oculta con la función *tansig*
- Capa de salida lineal

Cada capa, excepto la capa de entrada tiene una matriz de peso W , un vector de polarización b y un vector de salida a . Las matrices de peso conectadas a las entradas denominadas ponderaciones de entrada (IW) y las matrices de ponderación procedentes de salidas de capa ocultas denominadas ponderaciones de capa (LW). Además, los superíndices se usan para denotar la fuente (segundo índice) y el destino (primer índice) para los diversos pesos y otros elementos de la red, p es el vector de entrada, n es la salida de capa antes de la función de transferencia y a es el vector de salida real de una capa.

Dado que no existe una forma específica de averiguar el número de neuronas ocultas, se ha determinado mediante el método de prueba y error, es decir, cuando se trabaja con la red neuronal se hicieron alrededor de 74 pruebas para probar la eficacia y eficiencia de la red, tratando de incrementar el porcentaje de confianza de las pruebas. Los vectores de características de entrada se normalizan de menos a más para que sean más eficientes y rápidos.

Como se vio en el tema anterior, la red para el entrenamiento que se usó es backpropagation, por lo que los pesos fueron asignados por el algoritmo para mejora del rendimiento (MSE – error cuadrático medio) utilizando el gradiente de la función de rendimiento. El algoritmo de Levenberg-Marquardt (trainlm) fue para el entrenamiento de propagación hacia atrás. Es el método más rápido para el entrenamiento de redes neuronales de avance moderado y se basa en técnicas de optimización numérica. Sin embargo, se probaron las 14 funciones de entrenamiento que permite Matlab con diferentes tipos de redes, en total se realizaron 174 pruebas.

Se probaron además menos entradas y se incrementaron las neuronas para las pruebas.

Se utilizaron dos condiciones de parada temprana: error cuadrático medio total, $MSE \leq 0.001$ o entrenamiento detenido después de 1000 períodos (epochs). Los pesos y el sesgo de la capa de entrada y la capa oculta se guardaron después de cada sesión de entrenamiento.

Cuando los resultados de la simulación no son satisfactorios, la red se entrena nuevamente.

Debido a lo anterior, es importante reconocer que las muestras con las que se cuenta sean apropiadas para el entrenamiento.

El objetivo principal del procesamiento de la señal de electromiografía (EMG) en el sistema modular es la aplicación clínica. Generalmente se utiliza para el diagnóstico de enfermedades neurológicas y problemas neuromusculares.

EMG está controlado por el sistema nervioso y depende de las propiedades anatómicas y psicológicas de los músculos. La EMG suele ser una función del tiempo, descrita en términos de amplitud, frecuencia y fase, como se explica en el capítulo 2 de este documento de tesis.

Existen diversos enfoques tanto en la adquisición de datos, preprocesamiento de los datos, modelado de datos, análisis y en la interpretación de estos.

4. Resultados de los Experimentos

4.1 Introducción

En el presente capítulo se evalúa la funcionalidad y la efectividad de los diferentes módulos que componen el sistema de reconocimiento de patrones para el diagnóstico de enfermedades neuromusculares a través de señales EMG. Con la ayuda de Matlab se importan los archivos de las señales validas y acondicionadas, las cuales están almacenadas en una base de datos para su acondicionamiento digital (filtrado paso banda), posteriormente y se grafican utilizando Matlab. A continuación, se extraen 19 características de las señales EMG previamente procesadas para su clasificación dónde finalmente se evalúa el desempeño de las RNAs propuestas y se selecciona la que obtuvo mejor precisión en los resultados del reconocimiento de patrones de enfermedades neuromusculares.

4.2 Base de datos

Se tiene una base de datos de más de 200 señales EMG, sin embargo de ese universo, solo 77 señales de diferentes pacientes son aptas para la clasificación por su origen (Physiology, 2018) (Dua & Graff, 2019) (Clinic, 2019) (National institute of neurological disorders, 2015) (IEEE data Port). Dichas señales EMG provienen de pacientes sanos, con miopatía y neuropatía, con diferentes edades (25 ± 65) y género. A continuación, se presenta la Tabla 7 y como se observa se presenta las diferentes bases de datos obtenidas de diferentes bibliografías.

Base de datos	Diagnóstico	Músculo	Estado	Pacientes (señales)
(Dua & Graff, 2019)	Sano	Biceps_Femoral	Reposo	4
(Dua & Graff, 2019)	Sano	Semitendensioso	Reposo	4
(Dua & Graff, 2019)	Sano	Recto Femoral	Reposo	4
(Dua & Graff, 2019)	Sano	Vasto Medial	Reposo	4
(Physiology, 2018)	Sano	Tibialis Anterior	Reposo – ligera resistencia- Reposo	3
(Physiology, 2018)	Neuropatía	Tibialis Anterior	Reposo – ligera resistencia- Reposo	7
(Physiology, 2018)	Miopatía	Tibialis Anterior		6
(Dua & Graff, 2019)	Miopatía	Vasto Medial	Reposo	1

(Dua & Graff, 2019)	Mio/Neu	Biceps_Femoral	Reposo	11
(Dua & Graff, 2019)	Mio/Neu	Semitendinoso	Reposo	11
(Dua & Graff, 2019)	Mio/Neu	Recto Femoral	Reposo	11
(Dua & Graff, 2019)	Mio/Neu	Vasto Medial	Reposo	11
Adquisición propia	Sano	Biceps Braquial	Reposo	3
Adquisición propia	Sano	Biceps Braquial	Ligera contracción	3

Tabla 7. Diferentes tipos de señales EMG de la base de datos

La base de datos se conforma de esta manera pues el estado en que fue recopilada la señal es en estado de reposo, dónde el paciente no hace contracciones constantes de manera consiente y se puede analizar la señal mejor.

Las Figuras 21, 22 y 23 muestran las señales EMG de 3 personas con miopatía, neuropatía y persona sana respectivamente.

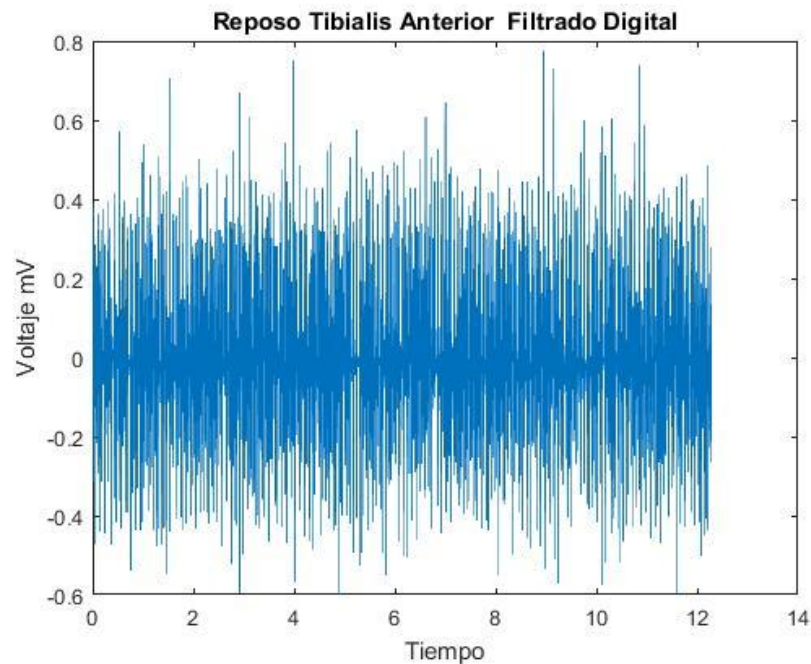


Figura 21. Señal EMG con miopatía en un tiempo de adquisición de 12 segundos

Como se puede observar en la Figura 21, es una señal de un sujeto diagnosticado con miopatía y no se alcanza a distinguir los potenciales de acción, por el exceso de ruido.

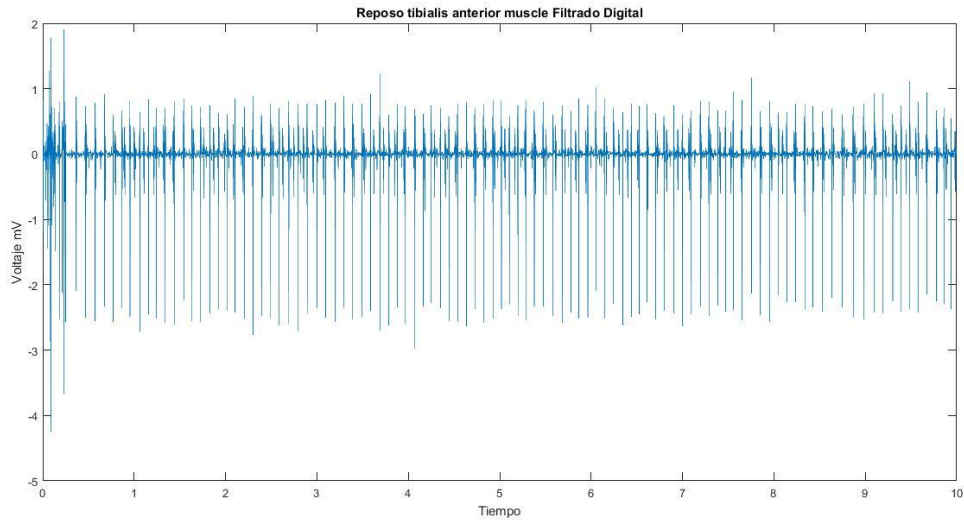


Figura 22. Señal EMG con neuropatía en un tiempo de 10 segundos

Como se puede observar en la Figura 22, es una señal de un sujeto diagnosticado con neuropatía a pesar del ruido se alcanza a distinguir los potenciales de acción clásicos de una neuropatía.

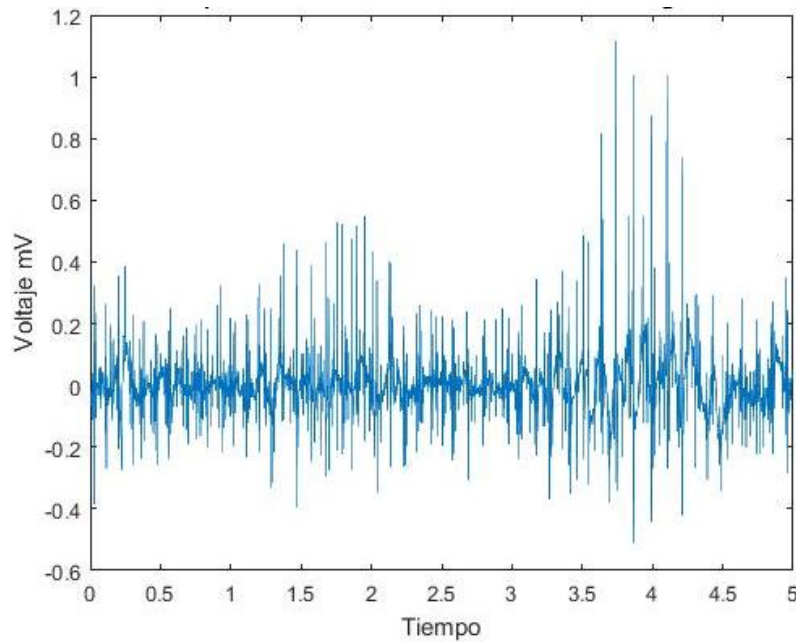


Figura 23. Señal EMG sana en un tiempo de 5 segundos

La Figura 23 muestra un ejemplo de la señal EMG sana, sin filtrado, sin embargo, puede observarse que el voltaje del mismo no supera el -0.5mV , por lo que aún con ruido la señal sana puede ser observada.

Para obtener la extracción de características de las señales EMG y la mejor clasificación de estas, se requiere primero acondicionar las señales con filtros digitales pasa banda de acuerdo con las recomendaciones propuestas en diferentes trabajos que toman en cuenta el tipo y la región del músculo dónde se miden. En la Tabla 8 se enlistan los trabajos revisados y los parámetros sugeridos por la international society of electrophysiology and kinesiology (ISEK, 2015) y por (William, 2019)

Músculo	f_{low}	f_{high}	Fs	Referencia
ta, pl, mg, so, tp	40	400, ~80	250	(Chen & Chen, 2020)
ta, mg	6	400	1000	(Chugani, 1998)
Bb	5	1000(2p Cheb)	2000	(Burden & R., 1999)
Bb	10	450(2p)	1024	(Rainoldi, Melchiorri, & Caruso, 2004) (A, y otros, 2000)
mg, ta	20	800	1000	(LaMontagne, Anthony, Keegel, & Tessa, 2012)
vl, vm, st, bf	20	450	1000	(A.M., M, & V, 2003)
vl, vm, st, bf, rf, mg	5.5	600	1000	(Benoit, M., G., & A., 2003)
mg, ta	16	600	1000	(Roetenberg, Buurke, Veltink, Forner Cordero, & Hermens, 2003)
VI	10	500	1000	(Solnik, y otros, 2008)
ra, d, lmf	10	1000	2000	(Tsao & Hodges, 2007)
st, sm, bf	5	10000	2500	(Onishi, y otros, 2002)
st, bf, gm, rf, ta, so, al	150	1000	2500	(Mulroy, 2003)

Tabla 8. Parámetros de muestreo recomendados para el acondicionamiento digital de las señales EMG de artículos de revistas seleccionados

La notación de los músculos es la siguiente:

al = aductor largo, bb = bíceps braquial, bf = bíceps femoral, d = deltoides, gm = glúteo mayor, lmf = multifidus lumbar, mg = gastrocnemio medial, ra = recto abdominal, rf = recto femoral, sm = semimembranoso, st = semitendinoso, ta = tibial anterior, tp = tibial posterior, vl = vasto lateral, vm = vasto medial

Todos los electrodos son electrodos de superficie excepto (Mulroy, 2003) y (Onishi, y otros, 2002), ambos de alambre fino y (Tsao & Hodges, 2007) de superficie y aguja de alambre fino. flow, fhigh = frecuencias bajas y altas del filtro de paso de banda analógico utilizado antes de muestrear la señal; fs = velocidad de muestreo. Los valores de flujo, fhigh y fs están en Hz. (William, 2019)

Después de haber realizado un filtrado en todas las señales, a continuación, se muestran las Figuras 24, 25 y 26, después de haberle hecho el procesamiento y filtrado digital.

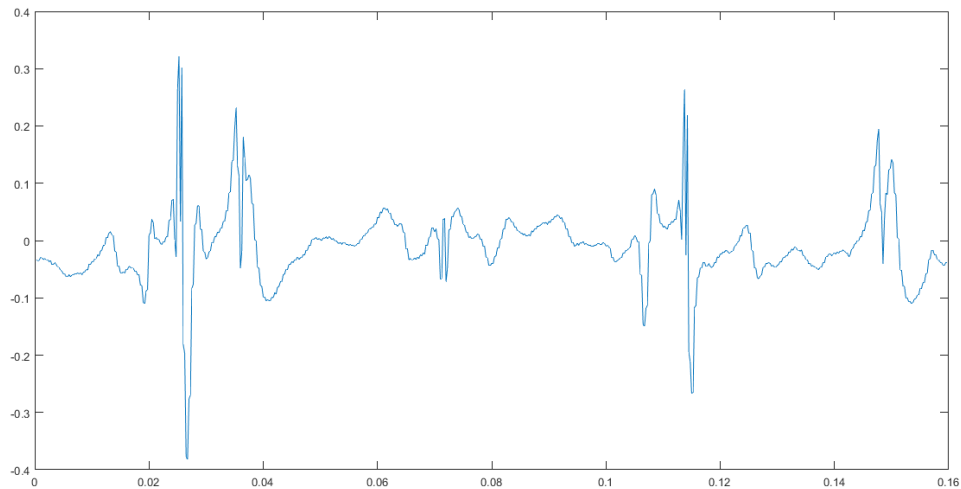


Figura 24. Señal EMG sana filtrada con filtros pasa banda

La Figura 24 se muestra la señal filtrada de la Figura 21, se hace una ventana de 0.16 segundos para un mejor análisis, y como se observa la forma de la onda de la señal la amplitud es normal pero el PAUM será de corta duración.

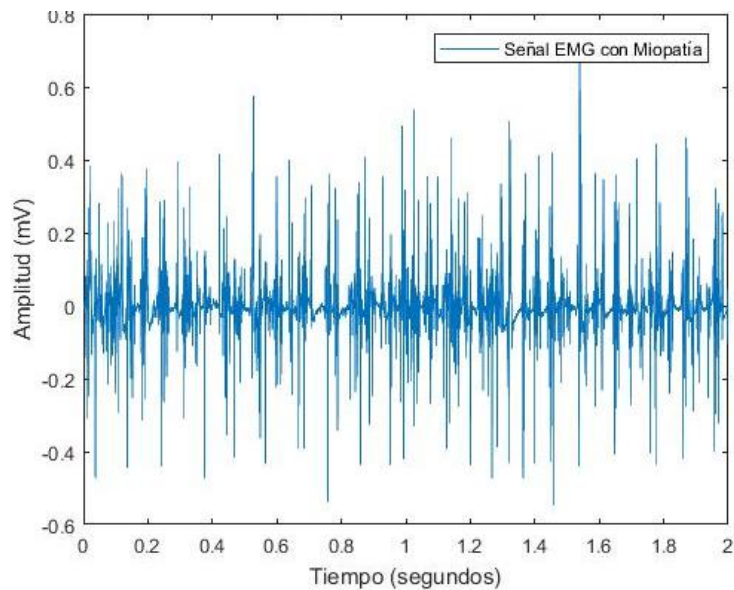


Figura 25. Señal EMG con miopatía filtrada

La Figura 25 muestra la señal filtrada de la Figura 22, se hace una ventana de 2 segundos para un mejor análisis, y como se observa la forma de la onda de la señal la amplitud está desde 0.1 a 0.7 mV en el lado positivo, en el lado en -0.58 mV en su punto más bajo. Se nota incluso a comparación de la señal sana que pareciera que la persona hizo contracciones o movimientos, sin embargo, la toma de la señal fue en completo estado de reposo.

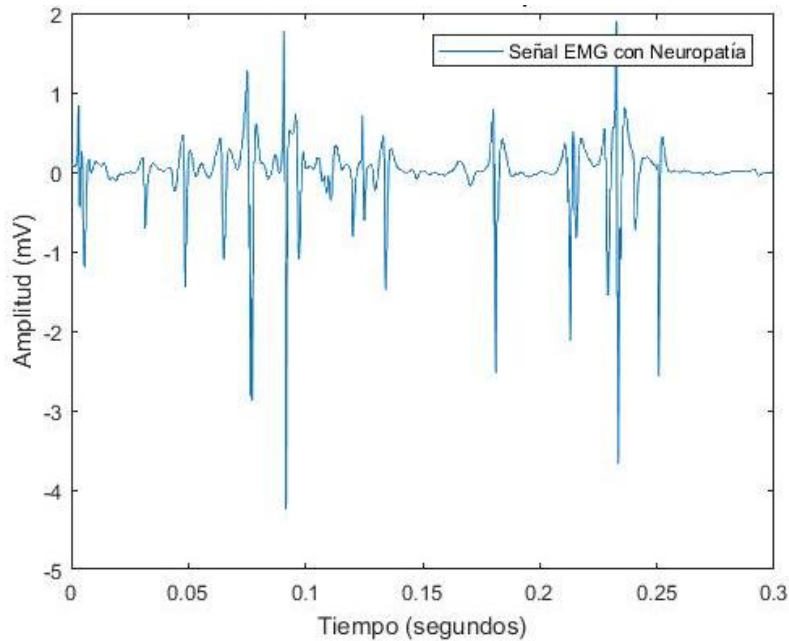


Figura 26. Señal EMG con neuropatía filtrada

La Figura 26 muestra la señal filtrada de la Figura 23, con una ventana de 0.3 segundos para un mejor análisis. Se puede notar que la amplitud en su parte negativa se encuentra entre -0.3 y -4.1, se observa la pérdida de las unidades motoras, las que quedan intentan reinervar las fibras musculares abandonadas, debido a ese incremento de densidad la amplitud y duración tiene un aumento.

4.3 Extracción de características

Para mejorar la precisión en la etapa de clasificación, 19 diferentes métodos de extracción de características son usados para obtener las entradas del entrenamiento mediante las RNAs.

A continuación, en la Tabla 9 se visualiza datos extraídos de los pacientes con miopatía, en la Tabla 10 los datos de pacientes con neuropatía. Se visualizan las características extraídas, en dónde:

- La primera fila indica cual el tipo de señal con la que se cuenta.
- P es el identificador del paciente con el número que lo representa.
- La columna de la extrema izquierda corresponde a las siglas pertenecientes a cada una de las características extraídas.

Target (Mio=2)	Miopatía							
	P1	P2	P3	P4	P5	P6	P7	P8
EMAV	0.036608	0.037054	0.036718	0.037614	0.035957	0.038145	0.005435	0.038116
EWL	2682.368513	1102.485614	1093.731831	1122.507933	1066.95677	782.291008	107.261047	2271.170697
MAV	0.006064	0.006061	0.006112	0.006382	0.005645	0.006497	0.000161	0.007493
WL	590.969572	240.485786	242.577121	253.273894	224.020289	177.329868	4.346737	593.685904
ZC	16749	6874	6691	7080	6412	4832	0	10064
SSC	18583	7624	7433	7839	7128	5358	0	10692
RMS	0.011031	0.010839	0.011403	0.012639	0.010446	0.012128	0.00023	0.018719
DAMV	0.012032	0.012024	0.012129	0.012664	0.011201	0.012894	0.000318	0.014842
DASDV	0.021883	0.021503	0.022619	0.025088	0.020726	0.02406	0.000455	0.037081
LD	0.002491	0.002503	0.00244	0.002627	0.002263	0.002619	0.000099	0.002176
MMAV1	0.004598	0.004542	0.004537	0.00482	0.004146	0.004932	0.000121	0.005528
MMAV2	0.00314	0.002879	0.002964	0.003151	0.002595	0.003423	0.000077	0.003551
MYOP	0.089153	0.09135	0.0897	0.091055	0.07995	0.099033	0	0.12795
SSI	5.976443	2.349555	2.600543	3.194605	2.182185	2.022765	0.000726	14.016098
VAR	0.000122	0.000117	0.00013	0.00016	0.000109	0.000147	0	0.00035
WAMP	16759	6877	6696	7087	6414	4837	0	10173
MFL	1.57893	1.112169	1.162748	1.266341	1.075348	1.037247	-2.9344	2.003667
MDF	2.908455	2.908901	2.905824	2.921664	2.913495	2.905755	2.81806	2.877658
MNF	2.914274	2.913703	2.91243	2.924728	2.917005	2.913555	2.830905	2.884285

Tabla 9. Extracción de características de 8 sujetos con miopatía

Target (Neu =2)	Neuropatía						
	P9	P10	P11	P12	P13	P14	P15
EMAV	0.038914	0.037325	0.076743	0.094665	0.036898	0.03843	0.03843
EWL	1157.726437	1111.46184	2351.020318	1602.4141	223.617561	1143.946303	1143.946303
MAV	0.007784	0.007202	0.025676	0.031352	0.007154	0.00756	0.00756
WL	308.34276	285.344696	1018.264971	696.135805	57.510854	299.63037	299.63037
ZC	5175	4889	11157	7871	1035	5091	5091
SSC	5491	5201	11632	8214	1101	5426	5426
RMS	0.019746	0.017632	0.057022	0.062007	0.016921	0.021317	0.021317
DAMV	0.015417	0.014267	0.050913	0.062188	0.014186	0.014982	0.014982
DASDV	0.039109	0.034937	0.113071	0.123015	0.033523	0.042227	0.042227
LD	0.002215	0.002137	0.007142	0.011422	0.002094	0.002238	0.002238
MMAV1	0.005688	0.005344	0.020609	0.023579	0.005406	0.005605	0.005605
MMAV2	0.003631	0.003493	0.011928	0.015655	0.003816	0.003738	0.003738
MYOP	0.13355	0.12235	0.3605	0.442291	0.125802	0.12645	0.12645
SSI	7.798316	6.217787	65.029151	43.039251	1.160786	9.088375	9.088375
VAR	0.00039	0.000311	0.003252	0.003845	0.000286	0.000454	0.000454
WAMP	5234	4939	11445	8069	1052	5136	5136
MFL	1.710326	1.597511	2.771975	2.566073	0.758083	1.787034	1.787034
MDF	2.875117	2.878465	2.901482	2.911931	2.875839	2.878989	2.878989
MNF	2.880638	2.888861	2.904036	2.91347	2.883835	2.884153	2.884153

Tabla 10. Extracción de características de 7 sujetos con neuropatía

La Tabla 11 muestra el resultado de los pacientes sanos mientras que la Tabla 12 muestra los datos de los pacientes con que han sido diagnosticados con miopatía y neuropatía, es decir han desarrollado las dos condiciones, en dónde:

- La primera fila corresponde a las siglas pertenecientes a cada una de las características extraídas.
- P es el identificador del paciente con el número que lo representa.

Target (Nor =1)	EMAV	EWL	MAV	WL	ZC	SSC	RMS	DAMV	DASDV	LD	MMAV1	MMAV2	MYOP	SSI	VAR	WAMP	MFL	MDF	MNF
P16	0.024757	733.849871	0.00312	123.850823	2984	3387	0.006566	0.006193	0.013025	0.001298	0.002358	0.001467	0.0304	0.862205	0.000043	2986	0.610859	2.909478	2.908311
P17	0.023475	693.503959	0.00264	104.703607	2735	3140	0.00488	0.005235	0.009673	0.001237	0.001996	0.001367	0.01695	0.476288	0.000024	2735	0.313317	2.893354	2.898417
P18	0.004165	91.80227	0.000094	2.856835	0	0	0.000118	0.000187	0.000233	0.000063	0.000071	0.000048	0	0.000212	0	0	-3.546894	2.87877	2.88319
P19	0.003856	84.989574	0.000083	2.517872	0	0	0.000104	0.000165	0.000206	0.000055	0.000062	0.000041	0	0.000165	0	0	-3.671084	2.899708	2.896554
P20	0.005103	112.696766	0.000145	4.366873	0	0	0.000193	0.000285	0.00038	0.000091	0.000109	0.000072	0	0.00057	0	0	-3.056865	2.807612	2.81536
P21	0.005058	111.404244	0.000138	4.161699	0	0	0.000181	0.000272	0.000356	0.000089	0.000102	0.000065	0	0.000499	0	0	-3.123864	2.803426	2.814084
P22	0.004165	91.80227	0.000094	2.856835	0	0	0.000118	0.000187	0.000233	0.000063	0.000071	0.000048	0	0.000212	0	0	-3.546894	2.87877	2.88319
P23	0.003856	84.989574	0.000083	2.517872	0	0	0.000104	0.000165	0.000206	0.000055	0.000062	0.000041	0	0.000165	0	0	-3.671084	2.899708	2.896554
P24	0.005103	112.696766	0.000145	4.366873	0	0	0.000193	0.000285	0.00038	0.000091	0.000109	0.000072	0	0.00057	0	0	-3.056865	2.807612	2.81536
P25	0.005058	111.404244	0.000138	4.161699	0	0	0.000181	0.000272	0.000356	0.000089	0.000102	0.000065	0	0.000499	0	0	-3.123864	2.803426	2.814084
P26	0.004352	128.575469	0.000104	4.218346	0	0	0.000131	0.000206	0.000258	0.000069	0.000078	0.000052	0	0.000349	0	0	-3.297122	2.863336	2.877988
P27	0.005099	150.568926	0.000142	5.730833	0	0	0.000187	0.00028	0.000369	0.000092	0.000107	0.000073	0	0.000719	0	0	-2.940833	2.799948	2.815815
P28	0.005626	166.480771	0.000184	7.42425	0	0	0.000253	0.000362	0.000497	0.000112	0.000142	0.000102	0	0.001314	0	0	-2.642288	2.767987	2.783213
P29	0.00464	136.847886	0.000123	4.954725	0	0	0.000162	0.000242	0.000319	0.000078	0.000093	0.000063	0	0.000541	0	0	-3.086957	2.759676	2.785548
P30	0.012876	5126.3826	0.000785	424.31723	210	260	0.001046	0.001556	0.002078	0.000474	0.000593	0.00041	0	0.298231	0.000001	210	0.081642	2.94772	2.944427
P31	0.01248	4598.94237	0.000758	380.964864	185	232	0.001026	0.001501	0.002038	0.000437	0.00057	0.000386	0	0.267249	0.000001	185	0.026589	2.945929	2.942322
P32	0.003171	237.543052	0.00005	2.445363	0	0	0.000052	0.000015	0.000026	0.000049	0.000038	0.000025	0	0.000437	0	0	-4.567574	0.00001	0.183356
P33	0.003203	22.597608	0.000051	0.04587	0	0	0.000051	0	0.000016	0.00005	0.000038	0.000025	0	0.000428	0	0	-5.030964	0.00001	0.073858
P34	0.017213	1923.61986	0.001566	235.058578	4366	4778	0.002537	0.003116	0.005054	0.000828	0.001234	0.000807	0.000093	0.485492	0.000006	4366	0.327935	2.999236	2.990959

Tabla 11. Extracción de características de 19 sujetos sanos

Target (Mio y Neu=2)	EMAV	EWL	MAV	WL	ZC	SSC	RMS	DAMV	DASDV	LD	MMAV1	MMAV2	MYOP	SSI	VAR	WAMP	MFL	MDF	MNF
P35	0.013097	257.301304	0.000828	22.174368	226	291	0.001553	0.001621	0.003039	0.00025	0.000529	0.000154	0	0.033	0.000002	226	-1.034654	2.744238	2.731782
P36	0.018394	366.609073	0.001737	46.726303	1482	1827	0.00303	0.003416	0.005958	0.000454	0.001215	0.000651	0.000731	0.125605	0.000009	1482	-0.361329	2.788802	2.781836
P37	0.005435	107.261047	0.000161	4.346737	0	0	0.00023	0.000318	0.000455	0.000099	0.000121	0.000077	0	0.000726	0	0	-2.9344	2.81806	2.830905
P38	0.004709	93.029797	0.000122	3.297407	0	0	0.000154	0.000241	0.000305	0.00008	0.000091	0.00006	0	0.000326	0	0	-3.333994	2.847234	2.862166
P39	0.013097	257.301304	0.000828	22.174368	226	291	0.001553	0.001621	0.003039	0.00025	0.000529	0.000154	0	0.033	0.000002	226	-1.034654	2.744238	2.731782
P40	0.018394	366.609073	0.001737	46.726303	1482	1827	0.00303	0.003416	0.005958	0.000454	0.001215	0.000651	0.000731	0.125605	0.000009	1482	-0.361329	2.788802	2.781836
P41	0.005435	107.261047	0.000161	4.346737	0	0	0.00023	0.000318	0.000455	0.000099	0.000121	0.000077	0	0.000726	0	0	-2.9344	2.81806	2.830905
P42	0.004709	93.029797	0.000122	3.297407	0	0	0.000154	0.000241	0.000305	0.00008	0.000091	0.00006	0	0.000326	0	0	-3.333994	2.847234	2.862166
P43	0.00419	235.659418	0.000099	7.631819	0	0	0.000124	0.000196	0.000246	0.000065	0.000074	0.00005	0	0.0006	0	0	-3.025914	2.87714	2.888217
P44	0.004082	229.671767	0.000094	7.265767	0	0	0.000118	0.000186	0.000234	0.000062	0.000071	0.000048	0	0.000541	0	0	-3.076798	2.92045	2.91362
P45	0.004994	280.248371	0.000131	10.080105	0	0	0.000167	0.000259	0.00033	0.000086	0.000098	0.000067	0	0.001092	0	0	-2.73032	2.824438	2.842342
P46	0.004651	261.387007	0.000117	9.011757	0	0	0.000147	0.000231	0.000291	0.000077	0.000088	0.00006	0	0.000844	0	0	-2.855912	2.870249	2.881572
P47	0.005143	223.75364	0.000158	9.344021	0	0	0.000214	0.000311	0.000422	0.000098	0.000122	0.00008	0	0.00138	0	0	-2.614797	2.80748	2.815017
P48	0.006362	280.504826	0.000353	20.826783	66	93	0.000767	0.000693	0.0015	0.000126	0.000303	0.00025	0	0.017711	0.000001	66	-1.346382	2.732656	2.72937
P49	0.006979	307.721508	0.000361	21.374902	18	27	0.000622	0.000711	0.001224	0.000176	0.000296	0.000205	0	0.011653	0	18	-1.550102	2.787603	2.783901
P50	0.006661	290.722506	0.000293	17.304906	28	37	0.000559	0.000575	0.001097	0.000134	0.000225	0.000121	0	0.009397	0	28	-1.659466	2.776496	2.766498
P51	0.004303	156.731323	0.000102	5.114723	0	0	0.000128	0.000202	0.000254	0.000068	0.000077	0.000052	0	0.000416	0	0	-3.210461	2.865366	2.876749
P52	0.004164	151.678875	0.000096	4.800396	0	0	0.00012	0.00019	0.000238	0.000064	0.000072	0.000048	0	0.000363	0	0	-3.276674	2.910968	2.905399
P53	0.006368	232.445949	0.000223	11.087198	0	0	0.000312	0.000439	0.000614	0.000131	0.000169	0.000109	0	0.002452	0	0	-2.327634	2.803804	2.806195
P54	0.005155	187.93768	0.000144	7.177375	0	0	0.000189	0.000284	0.000372	0.000091	0.000108	0.000069	0	0.000902	0	0	-2.827104	2.810916	2.828245
P55	0.006788	677.934999	0.000256	34.801056	7	10	0.000436	0.000502	0.000853	0.000131	0.000189	0.000095	0	0.013192	0	7	-1.49275	2.735154	2.738337

P56	0.008762	878.07212	0.000417	56.957604	83	114	0.000752	0.000821	0.001479	0.000188	0.000305	0.000167	0	0.039253	0.000001	83	-0.942442	2.798012	2.788207
P57	0.008827	896.289845	0.00053	72.310835	47	66	0.000866	0.001042	0.001703	0.000232	0.000425	0.000301	0	0.052083	0.000001	47	-0.801499	2.791409	2.783324
P58	0.008937	911.63818	0.000651	88.635112	534	722	0.001225	0.001278	0.002402	0.000207	0.000534	0.00038	0	0.104099	0.000002	534	-0.457949	2.761637	2.754324
P59	0.004004	82.040566	0.000091	2.569164	0	0	0.000115	0.000181	0.000227	0.000061	0.000069	0.000045	0	0.000188	0	0	-3.608012	2.889507	2.883074
P60	0.005223	107.058258	0.000156	4.377343	0	0	0.000245	0.000308	0.000484	0.000091	0.000113	0.000067	0	0.000855	0	0	-2.853507	2.818656	2.821399
P61	0.007087	144.825696	0.000262	7.318902	0	0	0.000425	0.000515	0.000834	0.000136	0.000187	0.00015	0	0.002568	0	0	-2.308046	2.774403	2.768166
P62	0.005138	105.914374	0.000165	4.638688	0	0	0.000231	0.000327	0.000455	0.000101	0.000124	0.000082	0	0.000755	0	0	-2.915485	2.813436	2.82791
P63	0.004238	91.171498	0.0001	2.961305	0	0	0.000129	0.000199	0.000255	0.000065	0.000076	0.000052	0	0.000248	0	0	-3.468518	2.860317	2.874873
P64	0.004109	88.403861	0.000095	2.81062	0	0	0.00012	0.000188	0.000237	0.000063	0.000071	0.000049	0	0.000213	0	0	-3.54303	2.913732	2.908232
P65	0.005347	115.68947	0.000186	5.468402	0	0	0.000271	0.000367	0.000532	0.000108	0.000144	0.000119	0	0.001092	0	0	-2.73441	2.783732	2.789449
P66	0.004464	96.242361	0.000116	3.415252	0	0	0.000146	0.000229	0.000288	0.000076	0.000088	0.00006	0	0.000317	0	0	-3.346199	2.876433	2.88547
P67	0.007921	213.581965	0.000335	12.306934	6	9	0.000584	0.000658	0.001147	0.000153	0.000253	0.000149	0	0.006377	0	6	-1.85211	2.788456	2.777769
P68	0.004717	127.178493	0.000119	4.396084	0	0	0.000151	0.000235	0.000298	0.000078	0.000089	0.000058	0	0.000424	0	0	-3.201588	2.848745	2.862009
P69	0.007411	199.627865	0.00029	10.67591	0	0	0.000484	0.000571	0.000949	0.00014	0.00022	0.000131	0	0.004376	0	0	-2.041532	2.77076	2.767398
P70	0.007257	195.799151	0.000287	10.571182	0	0	0.000475	0.000565	0.000934	0.00014	0.000218	0.000125	0	0.004217	0	0	-2.057886	2.792653	2.787374
P71	0.006177	176.751018	0.000247	9.622333	0	0	0.000454	0.000487	0.000894	0.000118	0.000194	0.00015	0	0.004065	0	0	-2.074735	2.810949	2.805139
P72	0.004351	123.979673	0.000108	4.213955	0	0	0.000136	0.000213	0.00027	0.000072	0.000082	0.000055	0	0.000367	0	0	-3.272555	2.853573	2.869155
P73	0.006636	189.98825	0.000271	10.518251	0	0	0.000407	0.000533	0.0008	0.000145	0.00021	0.000154	0	0.00327	0	0	-2.185508	2.791692	2.78512
P74	0.00495	140.726215	0.000142	5.50465	0	0	0.000215	0.000279	0.000423	0.000083	0.000107	0.000078	0	0.000915	0	0	-2.822507	2.784562	2.787812
P75	0.00652	238.383225	0.000265	13.186244	0	0	0.000462	0.000521	0.000907	0.000119	0.000198	0.000145	0	0.005405	0	0	-1.936429	2.769484	2.7612
P76	0.005034	183.670621	0.00014	6.987275	0	0	0.000214	0.000276	0.000422	0.000083	0.000103	0.000061	0	0.001159	0	0	-2.701281	2.809385	2.821129
P77	0.00643	234.751521	0.000242	12.047198	0	0	0.000407	0.000476	0.000801	0.000119	0.000179	0.000131	0	0.004198	0	0	-2.060911	2.782986	2.781532
P78	0.00713	260.281159	0.000303	15.062905	1	2	0.00057	0.000595	0.00112	0.000129	0.000222	0.000153	0	0.008231	0	1	-1.725147	2.776115	2.771859

Tabla 12. Extracción de características de 44 sujetos con miopatía y neuropatía (ambos diagnósticos)

A pesar de que los datos de las muestras con las que se cuentan algunos son de miopatía y otros de neuropatía, se consideraron todos en un mismo caso (enfermos), es decir, se agrupan en una misma clase, ya que como no se contaban con más casos enfermos se tomó esa decisión. Y los datos de las muestras sanas, se clasifican en otra clase, por lo tanto, se tienen dos clases para el clasificador:

- Normales (21 muestras sanas)
- Enfermos (59 muestras con miopatía y neuropatía)

4.4 Clasificador

Inmediatamente después extraídas las características de cada sujeto, las mismas se seleccionan de manera aleatoria para ingreso a la red neuronal.

Los experimentos con los diferentes tipos de redes neuronales en Matlab y sus diferentes combinaciones se muestran a continuación:

Tipo de red neuronal	
1	Cascade Forward Backprop
2	Competitive
3	Elman Backprop
4	Feed-forward backprop
5	Feed-forward distributed time delay
6	Feed-forward time delay
7	Generalized regression
8	Hopfield
9	Layer Recurrent
10	Linear layer (design)
11	Linear layer (Train)
12	LVQ
13	NARX
14	NARX Series-Parallel
15	Perceptron
16	Probabilistic
17	Radial Basis (Exact Fit)
18	Radial Basis (Fewer Neurons)
19	Self-organization map

Tabla 13. Tipo de redes neuronales

De la Tabla 13 se eligieron solo las redes 1,4,15 y 18 (cascade forward backprop, feed-forward backprop, probabilistic, perceptron y radial basis) para hacer los experimentos con diferentes combinaciones y optimizaciones para obtener mejores resultados. La elección dada fue debido a las referencias vigentes en este documento de tesis.

(Subasi, Abdulhamit, 2019) Dedicó el capítulo 3 para las técnicas de procesamiento de bioseñales en condiciones patológicas y diferentes técnicas de aprendizaje automático, incluidas redes neuronales basadas en retroalimentación.

(Toledo, Rodríguez, Gómez, & Jauregui, 2019) Hace un breve análisis en la página 5 de los diferentes clasificadores para movimientos de mano, con diferentes características extraídas y las agrupa.

(MATLAB, 2004) En el capítulo 5 backpropagation, brinda una amplia explicación del uso de la red, sus ventajas y desventajas, y las mejores combinaciones para una mejor precisión.

Las funciones de entrenamiento son básicamente algoritmos de entrenamiento que aplican reglas de aprendizaje de peso y sesgo, se utiliza para entrenar a la red neuronal para que reconozca una determinada entrada y la asigne a una salida. Crea un procedimiento paso a paso para ajustar los pesos de la conexión de la red neuronal artificial.

En el entrenamiento supervisado, la salida deseada para cada vector de entrada de un conjunto de entrenamiento se presenta a la red y pueden ser necesarias muchas iteraciones a través de los datos de entrenamiento para ajustar los pesos.

En el entrenamiento no supervisado, los pesos se ajustan sin especificar la salida correcta para ninguno de los vectores de entrada. (Demuth & Mark, 2004)

Para cada red neuronal creada, se utilizaron solo 4 funciones de entrenamiento (TrainBFG, TrainBR, TrainCGB, TrainCGF) de las 14 funciones disponibles como se visualiza en la Tabla 14. El criterio por el cual fueron elegidas fue de acuerdo con diferentes rendimientos con diferentes funciones de entrenamiento previamente analizados por (Vacic, 2005).

Función de entrenamiento	
1	TrainBFG
2	TrainBF
3	TrainCGB
4	TrainCGF
5	TrainCGP
6	TrainGD
7	TrainGDM
8	TrainGDA
9	TrainGDX
10	TrainM
11	TrainOSS
12	TrainR
13	TrainRP
14	TrainSCG

Tabla 14. Funciones de entrenamiento

La función de rendimiento por defecto en todas las configuraciones de las pruebas es el error cuadrático medio por sus siglas en inglés MSE. La Tabla 15 muestra las funciones de aprendizaje (proceso de presentar los patrones a aprender, a la red y el cambio de los pesos de las conexiones sinápticas usando una regla de aprendizaje) por defecto, en el caso de la red neuronal de tipo backpropagation, la función de aprendizaje sugerido por la bibliografía es LearnGDM. (Demuth & Mark, 2004)

Función de aprendizaje adaptado	
1	LearnGD
2	LearnGDM

Tabla 15. Funciones de aprendizaje de adaptación

Es muy importante poder calcular las derivadas de cualquier función de transferencia utilizada, la Tabla 16 muestra las diferentes funciones de transferencia de acuerdo con el número de neuronas utilizadas para generar la salida, generalmente en las redes multicapa utilizan LogSig o TanSig, y para redes backpropagation esas funciones son ideales. Para este proyecto de tesis se necesita precisión.

Función de transferencia	
1	LogSig
2	PureLIN
3	TanSig

Tabla 16. Función de transferencia

Las redes neuronales 1,4,15 y 18 de la Tabla 13 anteriores se combinaron con las 14 funciones de entrenamiento, las 2 funciones de aprendizaje y 3 de transferencia. Al combinar los parámetros disponibles de las RNAs se obtuvieron 174 pruebas, las cuales se describen en la Tabla 17:

	Network Type	Training Function	Performance Function	Transfer Function
Prueba 1	Cascade Forward Backprop	TrainBFG	LearnGD	LogSig
Prueba 2	Cascade Forward Backprop	TrainBFG	LearnGD	PureLIN
Prueba 3	Cascade Forward Backprop	TrainBFG	LearnGD	TanSIG
Prueba 4	Cascade Forward Backprop	TrainBFG	LearnGDM	LogSig
Prueba 5	Cascade Forward Backprop	TrainBFG	LearnGDM	PureLIN
Prueba 6	Cascade Forward Backprop	TrainBFG	LearnGDM	TanSIG
Prueba 7	Cascade Forward Backprop	TrainBR	LearnGD	LogSig
Prueba 8	Cascade Forward Backprop	TrainBR	LearnGD	PureLIN
Prueba 9	Cascade Forward Backprop	TrainBR	LearnGD	TanSIG
Prueba 10	Cascade Forward Backprop	TrainBR	LearnGDM	LogSig
Prueba 11	Cascade Forward Backprop	TrainBR	LearnGDM	PureLIN
Prueba 12	Cascade Forward Backprop	TrainBR	LearnGDM	TanSIG
Prueba 13	Cascade Forward Backprop	TrainCGB	LearnGD	LogSig
Prueba 14	Cascade Forward Backprop	TrainCGB	LearnGD	PureLIN
Prueba 15	Cascade Forward Backprop	TrainCGB	LearnGD	TanSIG
Prueba 16	Cascade Forward Backprop	TrainCGB	LearnGDM	LogSig
Prueba 17	Cascade Forward Backprop	TrainCGB	LearnGDM	PureLIN
Prueba 18	Cascade Forward Backprop	TrainCGB	LearnGDM	TanSIG
Prueba 19	Cascade Forward Backprop	TrainCGF	LearnGD	LogSig
Prueba 20	Cascade Forward Backprop	TrainCGF	LearnGD	PureLIN
Prueba 21	Cascade Forward Backprop	TrainCGF	LearnGD	TanSIG
Prueba 22	Cascade Forward Backprop	TrainCGF	LearnGDM	LogSig
Prueba 23	Cascade Forward Backprop	TrainCGF	LearnGDM	PureLIN
Prueba 24	Cascade Forward Backprop	TrainCGF	LearnGDM	TanSIG
Prueba 25	Cascade Forward Backprop	TrainCGP	LearnGD	LogSig
Prueba 26	Cascade Forward Backprop	TrainCGP	LearnGD	PureLIN
Prueba 27	Cascade Forward Backprop	TrainCGP	LearnGD	TanSIG
Prueba 28	Cascade Forward Backprop	TrainCGP	LearnGDM	LogSig
Prueba 29	Cascade Forward Backprop	TrainCGP	LearnGDM	PureLIN
Prueba 30	Cascade Forward Backprop	TrainCGP	LearnGDM	TanSIG
Prueba 31	Cascade Forward Backprop	TrainGD	LearnGD	LogSig
Prueba 32	Cascade Forward Backprop	TrainGD	LearnGD	PureLIN
Prueba 33	Cascade Forward Backprop	TrainGD	LearnGD	TanSIG
Prueba 34	Cascade Forward Backprop	TrainGD	LearnGDM	LogSig

Prueba 35	Cascade Forward Backprop	TrainGD	LearnGDM	PureLIN
Prueba 36	Cascade Forward Backprop	TrainGD	LearnGDM	TanSIG
Prueba 37	Cascade Forward Backprop	TrainGDM	LearnGD	LogSig
Prueba 38	Cascade Forward Backprop	TrainGDM	LearnGD	PureLIN
Prueba 39	Cascade Forward Backprop	TrainGDM	LearnGD	TanSIG
Prueba 40	Cascade Forward Backprop	TrainGDM	LearnGDM	LogSig
Prueba 41	Cascade Forward Backprop	TrainGDM	LearnGDM	PureLIN
Prueba 42	Cascade Forward Backprop	TrainGDM	LearnGDM	TanSIG
Prueba 43	Cascade Forward Backprop	TrainGDA	LearnGD	LogSig
Prueba 44	Cascade Forward Backprop	TrainGDA	LearnGD	PureLIN
Prueba 45	Cascade Forward Backprop	TrainGDA	LearnGD	TanSIG
Prueba 46	Cascade Forward Backprop	TrainGDA	LearnGDM	LogSig
Prueba 47	Cascade Forward Backprop	TrainGDA	LearnGDM	PureLIN
Prueba 48	Cascade Forward Backprop	TrainGDA	LearnGDM	TanSIG
Prueba 49	Cascade Forward Backprop	TrainGDX	LearnGD	LogSig
Prueba 50	Cascade Forward Backprop	TrainGDX	LearnGD	PureLIN
Prueba 51	Cascade Forward Backprop	TrainGDX	LearnGD	TanSIG
Prueba 52	Cascade Forward Backprop	TrainGDX	LearnGDM	LogSig
Prueba 53	Cascade Forward Backprop	TrainGDX	LearnGDM	PureLIN
Prueba 54	Cascade Forward Backprop	TrainGDX	LearnGDM	TanSIG
Prueba 55	Cascade Forward Backprop	TrainLM	LearnGD	LogSig
Prueba 56	Cascade Forward Backprop	TrainLM	LearnGD	PureLIN
Prueba 57	Cascade Forward Backprop	TrainLM	LearnGD	TanSIG
Prueba 58	Cascade Forward Backprop	TrainLM	LearnGDM	LogSig
Prueba 59	Cascade Forward Backprop	TrainLM	LearnGDM	PureLIN
Prueba 60	Cascade Forward Backprop	TrainLM	LearnGDM	TanSIG
Prueba 61	Cascade Forward Backprop	TrainOSS	LearnGD	LogSig
Prueba 62	Cascade Forward Backprop	TrainOSS	LearnGD	PureLIN
Prueba 63	Cascade Forward Backprop	TrainOSS	LearnGD	TanSIG
Prueba 64	Cascade Forward Backprop	TrainOSS	LearnGDM	LogSig
Prueba 65	Cascade Forward Backprop	TrainOSS	LearnGDM	PureLIN
Prueba 66	Cascade Forward Backprop	TrainOSS	LearnGDM	TanSIG
Prueba 67	Cascade Forward Backprop	TrainR	LearnGD	LogSig
Prueba 68	Cascade Forward Backprop	TrainR	LearnGD	PureLIN
Prueba 69	Cascade Forward Backprop	TrainR	LearnGD	TanSIG
Prueba 70	Cascade Forward Backprop	TrainR	LearnGD	LogSig
Prueba 71	Cascade Forward Backprop	TrainR	LearnGD	PureLIN
Prueba 72	Cascade Forward Backprop	TrainR	LearnGD	TanSIG
Prueba 73	Cascade Forward Backprop	TrainRP	LearnGD	LogSig
Prueba 74	Cascade Forward Backprop	TrainRP	LearnGD	PureLIN

Prueba 75	Cascade Forward Backprop	TrainRP	LearnGD	TanSIG
Prueba 76	Cascade Forward Backprop	TrainRP	LearnGDM	LogSig
Prueba 77	Cascade Forward Backprop	TrainRP	LearnGDM	PureLIN
Prueba 78	Cascade Forward Backprop	TrainRP	LearnGDM	TanSIG
Prueba 79	Cascade Forward Backprop	TrainSCG	LearnGD	LogSig
Prueba 80	Cascade Forward Backprop	TrainSCG	LearnGD	PureLIN
Prueba 81	Cascade Forward Backprop	TrainSCG	LearnGD	TanSIG
Prueba 82	Cascade Forward Backprop	TrainSCG	LearnGDM	LogSig
Prueba 83	Cascade Forward Backprop	TrainSCG	LearnGDM	PureLIN
Prueba 84	Cascade Forward Backprop	TrainSCG	LearnGDM	TanSIG
Prueba 85	Feed-forward backprop	TrainBFG	LearnGD	LogSig
Prueba 86	Feed-forward backprop	TrainBFG	LearnGD	PureLIN
Prueba 87	Feed-forward backprop	TrainBFG	LearnGD	TanSIG
Prueba 88	Feed-forward backprop	TrainBFG	LearnGDM	LogSig
Prueba 89	Feed-forward backprop	TrainBFG	LearnGDM	PureLIN
Prueba 90	Feed-forward backprop	TrainBFG	LearnGDM	TanSIG
Prueba 91	Feed-forward backprop	TrainBR	LearnGD	LogSig
Prueba 92	Feed-forward backprop	TrainBR	LearnGD	PureLIN
Prueba 93	Feed-forward backprop	TrainBR	LearnGD	TanSIG
Prueba 94	Feed-forward backprop	TrainBR	LearnGDM	LogSig
Prueba 95	Feed-forward backprop	TrainBR	LearnGDM	PureLIN
Prueba 96	Feed-forward backprop	TrainBR	LearnGDM	TanSIG
Prueba 97	Feed-forward backprop	TrainCGB	LearnGD	LogSig
Prueba 98	Feed-forward backprop	TrainCGB	LearnGD	PureLIN
Prueba 99	Feed-forward backprop	TrainCGB	LearnGD	TanSIG
Prueba 100	Feed-forward backprop	TrainCGB	LearnGDM	LogSig
Prueba 101	Feed-forward backprop	TrainCGB	LearnGDM	PureLIN
Prueba 102	Feed-forward backprop	TrainCGB	LearnGDM	TanSIG
Prueba 103	Feed-forward backprop	TrainCGF	LearnGD	LogSig
Prueba 104	Feed-forward backprop	TrainCGF	LearnGD	PureLIN
Prueba 105	Feed-forward backprop	TrainCGF	LearnGD	TanSIG
Prueba 106	Feed-forward backprop	TrainCGF	LearnGDM	LogSig
Prueba 107	Feed-forward backprop	TrainCGF	LearnGDM	PureLIN
Prueba 108	Feed-forward backprop	TrainCGF	LearnGDM	TanSIG
Prueba 109	Feed-forward backprop	TrainCGP	LearnGD	LogSig
Prueba 110	Feed-forward backprop	TrainCGP	LearnGD	PureLIN
Prueba 111	Feed-forward backprop	TrainCGP	LearnGD	TanSIG
Prueba 112	Feed-forward backprop	TrainCGP	LearnGDM	LogSig
Prueba 113	Feed-forward backprop	TrainCGP	LearnGDM	PureLIN
Prueba 114	Feed-forward backprop	TrainCGP	LearnGDM	TanSIG

Prueba 115	Feed-forward backprop	TrainGD	LearnGD	LogSig
Prueba 116	Feed-forward backprop	TrainGD	LearnGD	PureLIN
Prueba 117	Feed-forward backprop	TrainGD	LearnGD	TanSIG
Prueba 118	Feed-forward backprop	TrainGD	LearnGDM	LogSig
Prueba 119	Feed-forward backprop	TrainGD	LearnGDM	PureLIN
Prueba 120	Feed-forward backprop	TrainGD	LearnGDM	TanSIG
Prueba 121	Feed-forward backprop	TrainGDM	LearnGD	LogSig
Prueba 122	Feed-forward backprop	TrainGDM	LearnGD	PureLIN
Prueba 123	Feed-forward backprop	TrainGDM	LearnGD	TanSIG
Prueba 124	Feed-forward backprop	TrainGDM	LearnGDM	LogSig
Prueba 125	Feed-forward backprop	TrainGDM	LearnGDM	PureLIN
Prueba 126	Feed-forward backprop	TrainGDM	LearnGDM	TanSIG
Prueba 127	Feed-forward backprop	TrainGDA	LearnGD	LogSig
Prueba 128	Feed-forward backprop	TrainGDA	LearnGD	PureLIN
Prueba 129	Feed-forward backprop	TrainGDA	LearnGD	TanSIG
Prueba 130	Feed-forward backprop	TrainGDA	LearnGDM	LogSig
Prueba 131	Feed-forward backprop	TrainGDA	LearnGDM	PureLIN
Prueba 132	Feed-forward backprop	TrainGDA	LearnGDM	TanSIG
Prueba 133	Feed-forward backprop	TrainGDX	LearnGD	LogSig
Prueba 134	Feed-forward backprop	TrainGDX	LearnGD	PureLIN
Prueba 135	Feed-forward backprop	TrainGDX	LearnGD	TanSIG
Prueba 136	Feed-forward backprop	TrainGDX	LearnGDM	LogSig
Prueba 137	Feed-forward backprop	TrainGDX	LearnGDM	PureLIN
Prueba 138	Feed-forward backprop	TrainGDX	LearnGDM	TanSIG
Prueba 139	Feed-forward backprop	TrainM	LearnGD	LogSig
Prueba 140	Feed-forward backprop	TrainM	LearnGD	PureLIN
Prueba 141	Feed-forward backprop	TrainM	LearnGD	TanSIG
Prueba 142	Feed-forward backprop	TrainM	LearnGDM	LogSig
Prueba 143	Feed-forward backprop	TrainM	LearnGDM	PureLIN
Prueba 144	Feed-forward backprop	TrainM	LearnGDM	TanSIG
Prueba 145	Feed-forward backprop	TrainOSS	LearnGD	LogSig
Prueba 146	Feed-forward backprop	TrainOSS	LearnGD	PureLIN
Prueba 147	Feed-forward backprop	TrainOSS	LearnGD	TanSIG
Prueba 148	Feed-forward backprop	TrainOSS	LearnGDM	LogSig
Prueba 149	Feed-forward backprop	TrainOSS	LearnGDM	PureLIN
Prueba 150	Feed-forward backprop	TrainOSS	LearnGDM	TanSIG
Prueba 151	Feed-forward backprop	TrainR	LearnGD	LogSig
Prueba 152	Feed-forward backprop	TrainR	LearnGD	PureLIN
Prueba 153	Feed-forward backprop	TrainR	LearnGD	TanSIG
Prueba 154	Feed-forward backprop	TrainR	LearnGDM	LogSig

Prueba 155	Feed-forward backprop	TrainR	LearnGDM	PureLIN
Prueba 156	Feed-forward backprop	TrainR	LearnGDM	TanSIG
Prueba 157	Feed-forward backprop	TrainRP	LearnGD	LogSig
Prueba 158	Feed-forward backprop	TrainRP	LearnGD	PureLIN
Prueba 159	Feed-forward backprop	TrainRP	LearnGD	TanSIG
Prueba 160	Feed-forward backprop	TrainRP	LearnGDM	LogSig
Prueba 161	Feed-forward backprop	TrainRP	LearnGDM	PureLIN
Prueba 162	Feed-forward backprop	TrainRP	LearnGDM	TanSIG
Prueba 163	Feed-forward backprop	TrainSCG	LearnGD	LogSig
Prueba 164	Feed-forward backprop	TrainSCG	LearnGD	PureLIN
Prueba 165	Feed-forward backprop	TrainSCG	LearnGD	TanSIG
Prueba 166	Feed-forward backprop	TrainSCG	LearnGDM	LogSig
Prueba 167	Feed-forward backprop	TrainSCG	LearnGDM	PureLIN
Prueba 168	Feed-forward backprop	TrainSCG	LearnGDM	TanSIG
Prueba 169	Perceptron	HARDLIM	LearnP	
Prueba 170	Perceptron	HARDLIM	LearnPN	
Prueba 171	Perceptron	HARDLIMS	LearnP	
Prueba 172	Perceptron	HARDLIMS	LearnPN	
Prueba 173	Probabilistic	Spread Constant: 1.0		
Prueba 174	Radial Basis (Fewer Neurons)	Performance Goal: 0.9	Spread Constant: 1.0	

Tabla 17. Pruebas que se desarrollaron para esta tesis combinando los parámetros de las RNAs disponibles

Los pesos fueron asignados por el algoritmo por mejora del rendimiento (MSE – error cuadrático medio) utilizando el gradiente de la función de rendimiento. El algoritmo Levenberg-Marquardt (trainlm) para entrenamiento de propagación hacia atrás, es el método más rápido para el entrenamiento de redes neuronales de avance moderado y se basa en técnicas de optimización numérica.

Se probaron 4 de las 14 funciones de entrenamiento que permite Matlab de acuerdo a la bibliografía (Vacic, 2005), con los diferentes tipos de redes neuronales de la Tabla 13, en total se realizaron 174 pruebas como una primera aproximación y ver el comportamiento y los resultados. Se probaron además menos entradas y se incrementaron las neuronas para las mismas.

Se tienen 3 conjuntos en una red neuronal:

- Conjunto de entrenamiento. Es la entrada del algoritmo que se usa para establecer los pesos de las diferentes redes.

- Conjunto de validación. Se utiliza para elegir el algoritmo adecuado para su problema y reducir el sobreajuste. Compara el rendimiento de los diferentes y elige el mejor (cuando existe sobreajuste no tiene buen rendimiento).
- Conjunto de pruebas. Es la última fase después de haber elegido el algoritmo de entrenamiento y sus parámetros, utiliza un conjunto de datos nuevo y comprueba si realiza lo que debe hacer la red (coherencia) sobre los datos cargados y los datos reales. (Exchange, 2015)

Para el clasificador 1 (reunidos en la Tabla 18), se usaron los datos de la Tabla 9,10,11 y 12 (78 muestras en total) se dividieron los datos en:

- 70% para entrenamiento, es decir 55 muestras (obtenidos de la Tabla 9, Tabla 10 y Tabla 11).
- 30% para validación y pruebas. (obtenidos en la Tabla 9, Tabla 10 y Tabla 11)
 - 20% para validación, es decir 16 muestras.
 - 10% para pruebas, es decir 8 muestras.

Del clasificador 1 (Tabla 18) derivan los clasificadores 2,3,4, y 5 (Tabla 19, Tabla 20, Tabla 21 y Tabla 23), cada una detallada en el pie de la tabla como se encuentra distribuida.

En el clasificador 2 (Tabla 19), se dividieron los datos en:

- 65% para entrenamiento, es decir 48 muestras.
- 30% para validación y pruebas, es decir 26 muestras.
 - 20% para validación, es decir 15 muestras.
 - 10% para pruebas, es decir 11 muestras.

En el clasificador 3 (Tabla 20), se dividieron los datos en:

- 80% para entrenamiento, es decir 37 muestras.
- 20% para validación y pruebas, es decir 15 muestras.
 - 10% para validación, es decir 10 muestras.
 - 10% para pruebas, es decir 5 muestras.

En el clasificador 4 (Tabla 21), se dividieron los datos en:

- 70% para entrenamiento, es decir 37 muestras.
- 30% para validación y pruebas, es decir 12 muestras.

- 10% para validación, es decir 8 muestras.
- 10% para pruebas, es decir 4 muestras.

En el clasificador 5 (Tabla 23), se dividieron los datos en:

- 60% para entrenamiento, es decir 14 muestras.
- 40% para validación y pruebas, es decir 8 muestras.
 - 20% para validación, es decir 4 muestras.
 - 20% para pruebas, es decir 4 muestras.

La selección de las muestras se tomó de manera arbitraria, se trató de tener un balance en la red. Cuando los resultados de la simulación no son satisfactorios, la red se entrena nuevamente. Se utilizaron dos condiciones de parada temprana: error cuadrático medio total, $MSE = 0.001$ o entrenamiento detenido después de 1000 epochs. Los pesos y el sesgo de la capa de entrada y la capa oculta se guardaron después de cada sesión de entrenamiento.

Debido a lo anterior, es importante reconocer que las muestras con las que se cuenta sean apropiadas para el entrenamiento. Los datos para entrenamiento se muestran en la Tabla 18, Tabla 19, Tabla 20 y Tabla 21, se divide en pacientes (columnas) con las características ya extraídas y organizadas en 19 filas. La última columna representa los datos objetivo (target).

Los datos objetivos (objetivo) se dividen en:

- (1) Sano
- (2) Miopatía, neuropatía, o, miopatía y neuropatía

0.035957	1066.95677	0.005645	224.020289	6412	7128	0.010446	0.011201	0.020726	0.002263	0.004146	0.002595	0.07995	2.182185	0.000109	6414	1.075348	2.913495	2.917005	2
0.038145	782.291008	0.006497	177.329868	4832	5358	0.012128	0.012894	0.02406	0.002619	0.004932	0.003423	0.099033	2.022765	0.000147	4837	1.037247	2.905755	2.913555	2
0.036898	223.617561	0.007154	57.510854	1035	1101	0.016921	0.014186	0.033523	0.002094	0.005406	0.003816	0.125802	1.160786	0.000286	1052	0.758083	2.875839	2.883835	2
0.03843	1143.9463	0.00756	299.63037	5091	5426	0.021317	0.014982	0.042227	0.002238	0.005605	0.003738	0.12645	9.088375	0.000454	5136	1.787034	2.878989	2.884153	2
0.03843	1143.9463	0.00756	299.63037	5091	5426	0.021317	0.014982	0.042227	0.002238	0.005605	0.003738	0.12645	9.088375	0.000454	5136	1.787034	2.878989	2.884153	2
0.013097	257.301304	0.000828	22.174368	226	291	0.001553	0.001621	0.003039	0.00025	0.000529	0.000154	0	0.033	0.000002	226	-1.034654	2.744238	2.731782	2
0.018394	366.609073	0.001737	46.726303	1482	1827	0.00303	0.003416	0.005958	0.000454	0.001215	0.000651	0.000731	0.125605	0.000009	1482	-0.361329	2.788802	2.781836	2
0.005435	107.261047	0.000161	4.346737	0	0	0.00023	0.000318	0.000455	0.000099	0.000121	0.000077	0	0.000726	0	0	-2.9344	2.81806	2.830905	2
0.004709	93.029797	0.000122	3.297407	0	0	0.000154	0.000241	0.000305	0.00008	0.000091	0.00006	0	0.000326	0	0	-3.333994	2.847234	2.862166	2
0.013097	257.301304	0.000828	22.174368	226	291	0.001553	0.001621	0.003039	0.00025	0.000529	0.000154	0	0.033	0.000002	226	-1.034654	2.744238	2.731782	2
0.018394	366.609073	0.001737	46.726303	1482	1827	0.00303	0.003416	0.005958	0.000454	0.001215	0.000651	0.000731	0.125605	0.000009	1482	-0.361329	2.788802	2.781836	2
0.005435	107.261047	0.000161	4.346737	0	0	0.00023	0.000318	0.000455	0.000099	0.000121	0.000077	0	0.000726	0	0	-2.9344	2.81806	2.830905	2
0.004709	93.029797	0.000122	3.297407	0	0	0.000154	0.000241	0.000305	0.00008	0.000091	0.00006	0	0.000326	0	0	-3.333994	2.847234	2.862166	2
0.00419	235.659418	0.000099	7.631819	0	0	0.000124	0.000196	0.000246	0.000065	0.000074	0.00005	0	0.0006	0	0	-3.025914	2.87714	2.888217	2
0.004082	229.671767	0.000094	7.265767	0	0	0.000118	0.000186	0.000234	0.000062	0.000071	0.000048	0	0.000541	0	0	-3.076798	2.92045	2.91362	2
0.004994	280.248371	0.000131	10.080105	0	0	0.000167	0.000259	0.00033	0.000086	0.000098	0.000067	0	0.001092	0	0	-2.73032	2.824438	2.842342	2
0.004651	261.387007	0.000117	9.011757	0	0	0.000147	0.000231	0.000291	0.000077	0.000088	0.00006	0	0.000844	0	0	-2.855912	2.870249	2.881572	2
0.005143	223.75364	0.000158	9.344021	0	0	0.000214	0.000311	0.000422	0.000098	0.000122	0.00008	0	0.00138	0	0	-2.614797	2.80748	2.815017	2
0.006362	280.504826	0.000353	20.826783	66	93	0.000767	0.000693	0.0015	0.000126	0.000303	0.00025	0	0.017711	0.000001	66	-1.346382	2.732656	2.72937	2
0.006979	307.721508	0.000361	21.374902	18	27	0.000622	0.000711	0.001224	0.000176	0.000296	0.000205	0	0.011653	0	18	-1.550102	2.787603	2.783901	2
0.006661	290.722506	0.000293	17.304906	28	37	0.000559	0.000575	0.001097	0.000134	0.000225	0.000121	0	0.009397	0	28	-1.659466	2.776496	2.766498	2
0.004303	156.731323	0.000102	5.114723	0	0	0.000128	0.000202	0.000254	0.000068	0.000077	0.000052	0	0.000416	0	0	-3.210461	2.865366	2.876749	2
0.004164	151.678875	0.000096	4.800396	0	0	0.00012	0.00019	0.000238	0.000064	0.000072	0.000048	0	0.000363	0	0	-3.276674	2.910968	2.905399	2
0.006368	232.445949	0.000223	11.087198	0	0	0.000312	0.000439	0.000614	0.000131	0.000169	0.000109	0	0.002452	0	0	-2.327634	2.803804	2.806195	2
0.005155	187.93768	0.000144	7.177375	0	0	0.000189	0.000284	0.000372	0.000091	0.000108	0.000069	0	0.000902	0	0	-2.827104	2.810916	2.828245	2
0.006788	677.934999	0.000256	34.801056	7	10	0.000436	0.000502	0.000853	0.000131	0.000189	0.000095	0	0.013192	0	7	-1.49275	2.735154	2.738337	2
0.008762	878.07212	0.000417	56.957604	83	114	0.000752	0.000821	0.001479	0.000188	0.000305	0.000167	0	0.039253	0.000001	83	-0.942442	2.798012	2.788207	2
0.008827	896.289845	0.00053	72.310835	47	66	0.000866	0.001042	0.001703	0.000232	0.000425	0.000301	0	0.052083	0.000001	47	-0.801499	2.791409	2.783324	2

0.008937	911.63818	0.000651	88.635112	534	722	0.001225	0.001278	0.002402	0.000207	0.000534	0.00038	0	0.104099	0.000002	534	-0.457949	2.761637	2.754324	2
0.004004	82.040566	0.000091	2.569164	0	0	0.000115	0.000181	0.000227	0.000061	0.000069	0.000045	0	0.000188	0	0	-3.608012	2.889507	2.883074	2
0.005223	107.058258	0.000156	4.377343	0	0	0.000245	0.000308	0.000484	0.000091	0.000113	0.000067	0	0.000855	0	0	-2.853507	2.818656	2.821399	2
0.007087	144.825696	0.000262	7.318902	0	0	0.000425	0.000515	0.000834	0.000136	0.000187	0.00015	0	0.002568	0	0	-2.308046	2.774403	2.768166	2
0.005138	105.914374	0.000165	4.638688	0	0	0.000231	0.000327	0.000455	0.000101	0.000124	0.000082	0	0.000755	0	0	-2.915485	2.813436	2.82791	2
0.004238	91.171498	0.0001	2.961305	0	0	0.000129	0.000199	0.000255	0.000065	0.000076	0.000052	0	0.000248	0	0	-3.468518	2.860317	2.874873	2
0.004109	88.403861	0.000095	2.81062	0	0	0.00012	0.000188	0.000237	0.000063	0.000071	0.000049	0	0.000213	0	0	-3.54303	2.913732	2.908232	2
0.005347	115.68947	0.000186	5.468402	0	0	0.000271	0.000367	0.000532	0.000108	0.000144	0.000119	0	0.001092	0	0	-2.73441	2.783732	2.789449	2
0.004464	96.242361	0.000116	3.415252	0	0	0.000146	0.000229	0.000288	0.000076	0.000088	0.00006	0	0.000317	0	0	-3.346199	2.876433	2.88547	2
0.007921	213.581965	0.000335	12.306934	6	9	0.000584	0.000658	0.001147	0.000153	0.000253	0.000149	0	0.006377	0	6	-1.85211	2.788456	2.777769	2
0.004717	127.178493	0.000119	4.396084	0	0	0.000151	0.000235	0.000298	0.000078	0.000089	0.000058	0	0.000424	0	0	-3.201588	2.848745	2.862009	2
0.007411	199.627865	0.00029	10.67591	0	0	0.000484	0.000571	0.000949	0.00014	0.00022	0.000131	0	0.004376	0	0	-2.041532	2.77076	2.767398	2
0.007257	195.799151	0.000287	10.571182	0	0	0.000475	0.000565	0.000934	0.00014	0.000218	0.000125	0	0.004217	0	0	-2.057886	2.792653	2.787374	2
0.006177	176.751018	0.000247	9.622333	0	0	0.000454	0.000487	0.000894	0.000118	0.000194	0.00015	0	0.004065	0	0	-2.074735	2.810949	2.805139	2
0.004351	123.979673	0.000108	4.213955	0	0	0.000136	0.000213	0.00027	0.000072	0.000082	0.000055	0	0.000367	0	0	-3.272555	2.853573	2.869155	2
0.006636	189.98825	0.000271	10.518251	0	0	0.000407	0.000533	0.0008	0.000145	0.00021	0.000154	0	0.00327	0	0	-2.185508	2.791692	2.78512	2
0.00495	140.726215	0.000142	5.50465	0	0	0.000215	0.000279	0.000423	0.000083	0.000107	0.000078	0	0.000915	0	0	-2.822507	2.784562	2.787812	2
0.00652	238.383225	0.000265	13.186244	0	0	0.000462	0.000521	0.000907	0.000119	0.000198	0.000145	0	0.005405	0	0	-1.936429	2.769484	2.7612	2
0.005034	183.670621	0.00014	6.987275	0	0	0.000214	0.000276	0.000422	0.000083	0.000103	0.000061	0	0.001159	0	0	-2.701281	2.809385	2.821129	2
0.00643	234.751521	0.000242	12.047198	0	0	0.000407	0.000476	0.000801	0.000119	0.000179	0.000131	0	0.004198	0	0	-2.060911	2.782986	2.781532	2
0.00713	260.281159	0.000303	15.062905	1	2	0.00057	0.000595	0.00112	0.000129	0.000222	0.000153	0	0.008231	0	1	-1.725147	2.776115	2.771859	2
0.02669	25.126965	0.003515	4.455646	94	104	0.008134	0.006973	0.016136	0.001436	0.002545	0.002245	0.035994	0.042275	0.000066	94	-0.897511	2.905863	2.900756	1
0.024757	733.849871	0.00312	123.850823	2984	3387	0.006566	0.006193	0.013025	0.001298	0.002358	0.001467	0.0304	0.862205	0.000043	2986	0.610859	2.909478	2.908311	1
0.023475	693.503959	0.00264	104.703607	2735	3140	0.00488	0.005235	0.009673	0.001237	0.001996	0.001367	0.01695	0.476288	0.000024	2735	0.313317	2.893354	2.898417	1
0.036608	2682.36851	0.006064	590.969572	16749	18583	0.011031	0.012032	0.021883	0.002491	0.004598	0.00314	0.089153	5.976443	0.000122	16759	1.57893	2.908455	2.914274	2
0.037054	1102.48561	0.006061	240.485786	6874	7624	0.010839	0.012024	0.021503	0.002503	0.004542	0.002879	0.09135	2.349555	0.000117	6877	1.112169	2.908901	2.913703	2
0.036718	1093.73183	0.006112	242.577121	6691	7433	0.011403	0.012129	0.022619	0.00244	0.004537	0.002964	0.0897	2.600543	0.00013	6696	1.162748	2.905824	2.91243	2
0.037614	1122.50793	0.006382	253.273894	7080	7839	0.012639	0.012664	0.025088	0.002627	0.00482	0.003151	0.091055	3.194605	0.00016	7087	1.266341	2.921664	2.924728	2

0.005435	107.261047	0.000161	4.346737	0	0	0.00023	0.000318	0.000455	0.000099	0.000121	0.000077	0	0.000726	0	0	-2.9344	2.81806	2.830905	2
0.038116	2271.1707	0.007493	593.685904	10064	10692	0.018719	0.014842	0.037081	0.002176	0.005528	0.003551	0.12795	14.016098	0.00035	10173	2.003667	2.877658	2.884285	2
0.038914	1157.72644	0.007784	308.34276	5175	5491	0.019746	0.015417	0.039109	0.002215	0.005688	0.003631	0.13355	7.798316	0.00039	5234	1.710326	2.875117	2.880638	2
0.037325	1111.46184	0.007202	285.344696	4889	5201	0.017632	0.014267	0.034937	0.002137	0.005344	0.003493	0.12235	6.217787	0.000311	4939	1.597511	2.878465	2.888861	2
0.004165	91.80227	0.000094	2.856835	0	0	0.000118	0.000187	0.000233	0.000063	0.000071	0.000048	0	0.000212	0	0	-3.546894	2.87877	2.88319	1
0.003856	84.989574	0.000083	2.517872	0	0	0.000104	0.000165	0.000206	0.000055	0.000062	0.000041	0	0.000165	0	0	-3.671084	2.899708	2.896554	1
0.005103	112.696766	0.000145	4.366873	0	0	0.000193	0.000285	0.00038	0.000091	0.000109	0.000072	0	0.00057	0	0	-3.056865	2.807612	2.81536	1
0.005058	111.404244	0.000138	4.161699	0	0	0.000181	0.000272	0.000356	0.000089	0.000102	0.000065	0	0.000499	0	0	-3.123864	2.803426	2.814084	1
0.004165	91.80227	0.000094	2.856835	0	0	0.000118	0.000187	0.000233	0.000063	0.000071	0.000048	0	0.000212	0	0	-3.546894	2.87877	2.88319	1
0.003856	84.989574	0.000083	2.517872	0	0	0.000104	0.000165	0.000206	0.000055	0.000062	0.000041	0	0.000165	0	0	-3.671084	2.899708	2.896554	1
0.005103	112.696766	0.000145	4.366873	0	0	0.000193	0.000285	0.00038	0.000091	0.000109	0.000072	0	0.00057	0	0	-3.056865	2.807612	2.81536	1
0.005058	111.404244	0.000138	4.161699	0	0	0.000181	0.000272	0.000356	0.000089	0.000102	0.000065	0	0.000499	0	0	-3.123864	2.803426	2.814084	1
0.004352	128.575469	0.000104	4.218346	0	0	0.000131	0.000206	0.000258	0.000069	0.000078	0.000052	0	0.000349	0	0	-3.297122	2.863336	2.877988	1
0.005099	150.568926	0.000142	5.730833	0	0	0.000187	0.00028	0.000369	0.000092	0.000107	0.000073	0	0.000719	0	0	-2.940833	2.799948	2.815815	1
0.005626	166.480771	0.000184	7.42425	0	0	0.000253	0.000362	0.000497	0.000112	0.000142	0.000102	0	0.001314	0	0	-2.642288	2.767987	2.783213	1
0.00464	136.847886	0.000123	4.954725	0	0	0.000162	0.000242	0.000319	0.000078	0.000093	0.000063	0	0.000541	0	0	-3.086957	2.759676	2.785548	1
0.076743	2351.02032	0.025676	1018.26497	11157	11632	0.057022	0.050913	0.113071	0.007142	0.020609	0.011928	0.3605	65.029151	0.003252	11445	2.771975	2.901482	2.904036	2
0.094665	1602.4141	0.031352	696.135805	7871	8214	0.062007	0.062188	0.123015	0.011422	0.023579	0.015655	0.442291	43.039251	0.003845	8069	2.566073	2.911931	2.91347	2
0.012876	5126.3826	0.000785	424.31723	210	260	0.001046	0.001556	0.002078	0.000474	0.000593	0.00041	0	0.298231	0.000001	210	0.081642	2.94772	2.944427	1
0.01248	4598.942371	0.000758	380.964864	185	232	0.001026	0.001501	0.002038	0.000437	0.00057	0.000386	0	0.267249	0.000001	185	0.026589	2.945929	2.942322	1
0.003171	237.543052	0.00005	2.445363	0	0	0.000052	0.000015	0.000026	0.000049	0.000038	0.000025	0	0.000437	0	0	-4.567574	0.00001	0.183356	1
0.003203	22.597608	0.000051	0.04587	0	0	0.000051	0	0.000016	0.00005	0.000038	0.000025	0	0.000428	0	0	-5.030964	0.00001	0.073858	1
0.017213	1923.619857	0.001566	235.058578	4366	4778	0.002537	0.003116	0.005054	0.000828	0.001234	0.000807	0.000093	0.485492	0.000006	4366	0.327935	2.999236	2.990959	1

Tabla 18. Clasificador 1, con datos de entrada en la red neuronal representando a las columnas un total de 79 sujetos para entrenamiento (21 pacientes sanos y 59 pacientes enfermos)

0.035957	1066.95677	0.005645	224.020289	6412	7128	0.010446	0.011201	0.020726	0.002263	0.004146	0.002595	0.07995	2.182185	0.000109	6414	1.075348	2.913495	2.917005	2
0.038145	782.291008	0.006497	177.329868	4832	5358	0.012128	0.012894	0.02406	0.002619	0.004932	0.003423	0.099033	2.022765	0.000147	4837	1.037247	2.905755	2.913555	2
0.036898	223.617561	0.007154	57.510854	1035	1101	0.016921	0.014186	0.033523	0.002094	0.005406	0.003816	0.125802	1.160786	0.000286	1052	0.758083	2.875839	2.883835	2
0.03843	1143.9463	0.00756	299.63037	5091	5426	0.021317	0.014982	0.042227	0.002238	0.005605	0.003738	0.12645	9.088375	0.000454	5136	1.787034	2.878989	2.884153	2
0.03843	1143.9463	0.00756	299.63037	5091	5426	0.021317	0.014982	0.042227	0.002238	0.005605	0.003738	0.12645	9.088375	0.000454	5136	1.787034	2.878989	2.884153	2
0.013097	257.301304	0.000828	22.174368	226	291	0.001553	0.001621	0.003039	0.00025	0.000529	0.000154	0	0.033	0.000002	226	-1.034654	2.744238	2.731782	2
0.018394	366.609073	0.001737	46.726303	1482	1827	0.00303	0.003416	0.005958	0.000454	0.001215	0.000651	0.000731	0.125605	0.000009	1482	-0.361329	2.788802	2.781836	2
0.005435	107.261047	0.000161	4.346737	0	0	0.00023	0.000318	0.000455	0.000099	0.000121	0.000077	0	0.000726	0	0	-2.9344	2.81806	2.830905	2
0.004709	93.029797	0.000122	3.297407	0	0	0.000154	0.000241	0.000305	0.00008	0.000091	0.00006	0	0.000326	0	0	-3.333994	2.847234	2.862166	2
0.013097	257.301304	0.000828	22.174368	226	291	0.001553	0.001621	0.003039	0.00025	0.000529	0.000154	0	0.033	0.000002	226	-1.034654	2.744238	2.731782	2
0.018394	366.609073	0.001737	46.726303	1482	1827	0.00303	0.003416	0.005958	0.000454	0.001215	0.000651	0.000731	0.125605	0.000009	1482	-0.361329	2.788802	2.781836	2
0.005435	107.261047	0.000161	4.346737	0	0	0.00023	0.000318	0.000455	0.000099	0.000121	0.000077	0	0.000726	0	0	-2.9344	2.81806	2.830905	2
0.004709	93.029797	0.000122	3.297407	0	0	0.000154	0.000241	0.000305	0.00008	0.000091	0.00006	0	0.000326	0	0	-3.333994	2.847234	2.862166	2
0.00419	235.659418	0.000099	7.631819	0	0	0.000124	0.000196	0.000246	0.000065	0.000074	0.00005	0	0.0006	0	0	-3.025914	2.87714	2.888217	2
0.004082	229.671767	0.000094	7.265767	0	0	0.000118	0.000186	0.000234	0.000062	0.000071	0.000048	0	0.000541	0	0	-3.076798	2.92045	2.91362	2
0.004994	280.248371	0.000131	10.080105	0	0	0.000167	0.000259	0.00033	0.000086	0.000098	0.000067	0	0.001092	0	0	-2.73032	2.824438	2.842342	2
0.004651	261.387007	0.000117	9.011757	0	0	0.000147	0.000231	0.000291	0.000077	0.000088	0.00006	0	0.000844	0	0	-2.855912	2.870249	2.881572	2
0.005143	223.75364	0.000158	9.344021	0	0	0.000214	0.000311	0.000422	0.000098	0.000122	0.00008	0	0.00138	0	0	-2.614797	2.80748	2.815017	2
0.006362	280.504826	0.000353	20.826783	66	93	0.000767	0.000693	0.0015	0.000126	0.000303	0.00025	0	0.017711	0.000001	66	-1.346382	2.732656	2.72937	2
0.006979	307.721508	0.000361	21.374902	18	27	0.000622	0.000711	0.001224	0.000176	0.000296	0.000205	0	0.011653	0	18	-1.550102	2.787603	2.783901	2
0.006661	290.722506	0.000293	17.304906	28	37	0.000559	0.000575	0.001097	0.000134	0.000225	0.000121	0	0.009397	0	28	-1.659466	2.776496	2.766498	2
0.004303	156.731323	0.000102	5.114723	0	0	0.000128	0.000202	0.000254	0.000068	0.000077	0.000052	0	0.000416	0	0	-3.210461	2.865366	2.876749	2
0.004164	151.678875	0.000096	4.800396	0	0	0.00012	0.00019	0.000238	0.000064	0.000072	0.000048	0	0.000363	0	0	-3.276674	2.910968	2.905399	2
0.006368	232.445949	0.000223	11.087198	0	0	0.000312	0.000439	0.000614	0.000131	0.000169	0.000109	0	0.002452	0	0	-2.327634	2.803804	2.806195	2
0.005155	187.93768	0.000144	7.177375	0	0	0.000189	0.000284	0.000372	0.000091	0.000108	0.000069	0	0.000902	0	0	-2.827104	2.810916	2.828245	2
0.006788	677.934999	0.000256	34.801056	7	10	0.000436	0.000502	0.000853	0.000131	0.000189	0.000095	0	0.013192	0	7	-1.49275	2.735154	2.738337	2
0.008762	878.07212	0.000417	56.957604	83	114	0.000752	0.000821	0.001479	0.000188	0.000305	0.000167	0	0.039253	0.000001	83	-0.942442	2.798012	2.788207	2
0.008827	896.289845	0.00053	72.310835	47	66	0.000866	0.001042	0.001703	0.000232	0.000425	0.000301	0	0.052083	0.000001	47	-0.801499	2.791409	2.783324	2

0.008937	911.63818	0.000651	88.635112	534	722	0.001225	0.001278	0.002402	0.000207	0.000534	0.00038	0	0.104099	0.000002	534	-0.457949	2.761637	2.754324	2
0.004004	82.040566	0.000091	2.569164	0	0	0.000115	0.000181	0.000227	0.000061	0.000069	0.000045	0	0.000188	0	0	-3.608012	2.889507	2.883074	2
0.005223	107.058258	0.000156	4.377343	0	0	0.000245	0.000308	0.000484	0.000091	0.000113	0.000067	0	0.000855	0	0	-2.853507	2.818656	2.821399	2
0.007087	144.825696	0.000262	7.318902	0	0	0.000425	0.000515	0.000834	0.000136	0.000187	0.00015	0	0.002568	0	0	-2.308046	2.774403	2.768166	2
0.005138	105.914374	0.000165	4.638688	0	0	0.000231	0.000327	0.000455	0.000101	0.000124	0.000082	0	0.000755	0	0	-2.915485	2.813436	2.82791	2
0.004238	91.171498	0.0001	2.961305	0	0	0.000129	0.000199	0.000255	0.000065	0.000076	0.000052	0	0.000248	0	0	-3.468518	2.860317	2.874873	2
0.004109	88.403861	0.000095	2.81062	0	0	0.00012	0.000188	0.000237	0.000063	0.000071	0.000049	0	0.000213	0	0	-3.54303	2.913732	2.908232	2
0.005347	115.68947	0.000186	5.468402	0	0	0.000271	0.000367	0.000532	0.000108	0.000144	0.000119	0	0.001092	0	0	-2.73441	2.783732	2.789449	2
0.004464	96.242361	0.000116	3.415252	0	0	0.000146	0.000229	0.000288	0.000076	0.000088	0.00006	0	0.000317	0	0	-3.346199	2.876433	2.88547	2
0.007921	213.581965	0.000335	12.306934	6	9	0.000584	0.000658	0.001147	0.000153	0.000253	0.000149	0	0.006377	0	6	-1.85211	2.788456	2.777769	2
0.004717	127.178493	0.000119	4.396084	0	0	0.000151	0.000235	0.000298	0.000078	0.000089	0.000058	0	0.000424	0	0	-3.201588	2.848745	2.862009	2
0.007411	199.627865	0.00029	10.67591	0	0	0.000484	0.000571	0.000949	0.00014	0.00022	0.000131	0	0.004376	0	0	-2.041532	2.77076	2.767398	2
0.007257	195.799151	0.000287	10.571182	0	0	0.000475	0.000565	0.000934	0.00014	0.000218	0.000125	0	0.004217	0	0	-2.057886	2.792653	2.787374	2
0.006177	176.751018	0.000247	9.622333	0	0	0.000454	0.000487	0.000894	0.000118	0.000194	0.00015	0	0.004065	0	0	-2.074735	2.810949	2.805139	2
0.004351	123.979673	0.000108	4.213955	0	0	0.000136	0.000213	0.00027	0.000072	0.000082	0.000055	0	0.000367	0	0	-3.272555	2.853573	2.869155	2
0.006636	189.98825	0.000271	10.518251	0	0	0.000407	0.000533	0.0008	0.000145	0.00021	0.000154	0	0.00327	0	0	-2.185508	2.791692	2.78512	2
0.00495	140.726215	0.000142	5.50465	0	0	0.000215	0.000279	0.000423	0.000083	0.000107	0.000078	0	0.000915	0	0	-2.822507	2.784562	2.787812	2
0.00652	238.383225	0.000265	13.186244	0	0	0.000462	0.000521	0.000907	0.000119	0.000198	0.000145	0	0.005405	0	0	-1.936429	2.769484	2.7612	2
0.005034	183.670621	0.00014	6.987275	0	0	0.000214	0.000276	0.000422	0.000083	0.000103	0.000061	0	0.001159	0	0	-2.701281	2.809385	2.821129	2
0.00643	234.751521	0.000242	12.047198	0	0	0.000407	0.000476	0.000801	0.000119	0.000179	0.000131	0	0.004198	0	0	-2.060911	2.782986	2.781532	2
0.00713	260.281159	0.000303	15.062905	1	2	0.00057	0.000595	0.00112	0.000129	0.000222	0.000153	0	0.008231	0	1	-1.725147	2.776115	2.771859	2
0.02669	25.126965	0.003515	4.455646	94	104	0.008134	0.006973	0.016136	0.001436	0.002545	0.002245	0.035994	0.042275	0.000066	94	-0.897511	2.905863	2.900756	1
0.024757	733.849871	0.00312	123.850823	2984	3387	0.006566	0.006193	0.013025	0.001298	0.002358	0.001467	0.0304	0.862205	0.000043	2986	0.610859	2.909478	2.908311	1
0.023475	693.503959	0.00264	104.703607	2735	3140	0.00488	0.005235	0.009673	0.001237	0.001996	0.001367	0.01695	0.476288	0.000024	2735	0.313317	2.893354	2.898417	1
0.036608	2682.36851	0.006064	590.969572	16749	18583	0.011031	0.012032	0.021883	0.002491	0.004598	0.00314	0.089153	5.976443	0.000122	16759	1.57893	2.908455	2.914274	2
0.037054	1102.48561	0.006061	240.485786	6874	7624	0.010839	0.012024	0.021503	0.002503	0.004542	0.002879	0.09135	2.349555	0.000117	6877	1.112169	2.908901	2.913703	2
0.036718	1093.73183	0.006112	242.577121	6691	7433	0.011403	0.012129	0.022619	0.00244	0.004537	0.002964	0.0897	2.600543	0.00013	6696	1.162748	2.905824	2.91243	2
0.037614	1122.50793	0.006382	253.273894	7080	7839	0.012639	0.012664	0.025088	0.002627	0.00482	0.003151	0.091055	3.194605	0.00016	7087	1.266341	2.921664	2.924728	2

0.005435	107.261047	0.000161	4.346737	0	0	0.00023	0.000318	0.000455	0.000099	0.000121	0.000077	0	0.000726	0	0	-2.9344	2.81806	2.830905	2
0.038116	2271.1707	0.007493	593.685904	10064	10692	0.018719	0.014842	0.037081	0.002176	0.005528	0.003551	0.12795	14.016098	0.00035	10173	2.003667	2.877658	2.884285	2
0.038914	1157.72644	0.007784	308.34276	5175	5491	0.019746	0.015417	0.039109	0.002215	0.005688	0.003631	0.13355	7.798316	0.00039	5234	1.710326	2.875117	2.880638	2
0.037325	1111.46184	0.007202	285.344696	4889	5201	0.017632	0.014267	0.034937	0.002137	0.005344	0.003493	0.12235	6.217787	0.000311	4939	1.597511	2.878465	2.888861	2
0.004165	91.80227	0.000094	2.856835	0	0	0.000118	0.000187	0.000233	0.000063	0.000071	0.000048	0	0.000212	0	0	-3.546894	2.87877	2.88319	1
0.003856	84.989574	0.000083	2.517872	0	0	0.000104	0.000165	0.000206	0.000055	0.000062	0.000041	0	0.000165	0	0	-3.671084	2.899708	2.896554	1
0.005103	112.696766	0.000145	4.366873	0	0	0.000193	0.000285	0.00038	0.000091	0.000109	0.000072	0	0.00057	0	0	-3.056865	2.807612	2.81536	1
0.005058	111.404244	0.000138	4.161699	0	0	0.000181	0.000272	0.000356	0.000089	0.000102	0.000065	0	0.000499	0	0	-3.123864	2.803426	2.814084	1
0.004165	91.80227	0.000094	2.856835	0	0	0.000118	0.000187	0.000233	0.000063	0.000071	0.000048	0	0.000212	0	0	-3.546894	2.87877	2.88319	1
0.003856	84.989574	0.000083	2.517872	0	0	0.000104	0.000165	0.000206	0.000055	0.000062	0.000041	0	0.000165	0	0	-3.671084	2.899708	2.896554	1
0.005103	112.696766	0.000145	4.366873	0	0	0.000193	0.000285	0.00038	0.000091	0.000109	0.000072	0	0.00057	0	0	-3.056865	2.807612	2.81536	2
0.005058	111.404244	0.000138	4.161699	0	0	0.000181	0.000272	0.000356	0.000089	0.000102	0.000065	0	0.000499	0	0	-3.123864	2.803426	2.814084	1
0.004352	128.575469	0.000104	4.218346	0	0	0.000131	0.000206	0.000258	0.000069	0.000078	0.000052	0	0.000349	0	0	-3.297122	2.863336	2.877988	1
0.005099	150.568926	0.000142	5.730833	0	0	0.000187	0.00028	0.000369	0.000092	0.000107	0.000073	0	0.000719	0	0	-2.940833	2.799948	2.815815	1
0.005626	166.480771	0.000184	7.42425	0	0	0.000253	0.000362	0.000497	0.000112	0.000142	0.000102	0	0.001314	0	0	-2.642288	2.767987	2.783213	1
0.00464	136.847886	0.000123	4.954725	0	0	0.000162	0.000242	0.000319	0.000078	0.000093	0.000063	0	0.000541	0	0	-3.086957	2.759676	2.785548	1
0.076743	2351.02032	0.025676	1018.26497	11157	11632	0.057022	0.050913	0.113071	0.007142	0.020609	0.011928	0.3605	65.029151	0.003252	11445	2.771975	2.901482	2.904036	2
0.094665	1602.4141	0.031352	696.135805	7871	8214	0.062007	0.062188	0.123015	0.011422	0.023579	0.015655	0.442291	43.039251	0.003845	8069	2.566073	2.911931	2.91347	2

Tabla 19. Clasificador 2, con datos de entrada en la red neuronal representando a las columnas un total de 74 sujetos para entrenamiento (14 pacientes sanos y 59 pacientes enfermos)

0.035957	1066.95677	0.005645	224.020289	6412	7128	0.010446	0.011201	0.020726	0.002263	0.004146	0.002595	0.07995	2.182185	0.000109	6414	1.075348	2.913495	2.917005	2
0.038145	782.291008	0.006497	177.329868	4832	5358	0.012128	0.012894	0.02406	0.002619	0.004932	0.003423	0.099033	2.022765	0.000147	4837	1.037247	2.905755	2.913555	2
0.036898	223.617561	0.007154	57.510854	1035	1101	0.016921	0.014186	0.033523	0.002094	0.005406	0.003816	0.125802	1.160786	0.000286	1052	0.758083	2.875839	2.883835	2
0.03843	1143.9463	0.00756	299.63037	5091	5426	0.021317	0.014982	0.042227	0.002238	0.005605	0.003738	0.12645	9.088375	0.000454	5136	1.787034	2.878989	2.884153	2
0.03843	1143.9463	0.00756	299.63037	5091	5426	0.021317	0.014982	0.042227	0.002238	0.005605	0.003738	0.12645	9.088375	0.000454	5136	1.787034	2.878989	2.884153	2
0.013097	257.301304	0.000828	22.174368	226	291	0.001553	0.001621	0.003039	0.00025	0.000529	0.000154	0	0.033	0.000002	226	-1.034654	2.744238	2.731782	2
0.018394	366.609073	0.001737	46.726303	1482	1827	0.00303	0.003416	0.005958	0.000454	0.001215	0.000651	0.000731	0.125605	0.000009	1482	-0.361329	2.788802	2.781836	2
0.005435	107.261047	0.000161	4.346737	0	0	0.00023	0.000318	0.000455	0.000099	0.000121	0.000077	0	0.000726	0	0	-2.9344	2.81806	2.830905	2
0.004709	93.029797	0.000122	3.297407	0	0	0.000154	0.000241	0.000305	0.00008	0.000091	0.00006	0	0.000326	0	0	-3.333994	2.847234	2.862166	2
0.013097	257.301304	0.000828	22.174368	226	291	0.001553	0.001621	0.003039	0.00025	0.000529	0.000154	0	0.033	0.000002	226	-1.034654	2.744238	2.731782	2
0.018394	366.609073	0.001737	46.726303	1482	1827	0.00303	0.003416	0.005958	0.000454	0.001215	0.000651	0.000731	0.125605	0.000009	1482	-0.361329	2.788802	2.781836	2
0.005435	107.261047	0.000161	4.346737	0	0	0.00023	0.000318	0.000455	0.000099	0.000121	0.000077	0	0.000726	0	0	-2.9344	2.81806	2.830905	2
0.004709	93.029797	0.000122	3.297407	0	0	0.000154	0.000241	0.000305	0.00008	0.000091	0.00006	0	0.000326	0	0	-3.333994	2.847234	2.862166	2
0.00419	235.659418	0.000099	7.631819	0	0	0.000124	0.000196	0.000246	0.000065	0.000074	0.00005	0	0.0006	0	0	-3.025914	2.87714	2.888217	2
0.004082	229.671767	0.000094	7.265767	0	0	0.000118	0.000186	0.000234	0.000062	0.000071	0.000048	0	0.000541	0	0	-3.076798	2.92045	2.91362	2
0.004994	280.248371	0.000131	10.080105	0	0	0.000167	0.000259	0.00033	0.000086	0.000098	0.000067	0	0.001092	0	0	-2.73032	2.824438	2.842342	2
0.004651	261.387007	0.000117	9.011757	0	0	0.000147	0.000231	0.000291	0.000077	0.000088	0.00006	0	0.000844	0	0	-2.855912	2.870249	2.881572	2
0.005143	223.75364	0.000158	9.344021	0	0	0.000214	0.000311	0.000422	0.000098	0.000122	0.00008	0	0.00138	0	0	-2.614797	2.80748	2.815017	2
0.006362	280.504826	0.000353	20.826783	66	93	0.000767	0.000693	0.0015	0.000126	0.000303	0.00025	0	0.017711	0.000001	66	-1.346382	2.732656	2.72937	2
0.006979	307.721508	0.000361	21.374902	18	27	0.000622	0.000711	0.001224	0.000176	0.000296	0.000205	0	0.011653	0	18	-1.550102	2.787603	2.783901	2
0.006661	290.722506	0.000293	17.304906	28	37	0.000559	0.000575	0.001097	0.000134	0.000225	0.000121	0	0.009397	0	28	-1.659466	2.776496	2.766498	2
0.004303	156.731323	0.000102	5.114723	0	0	0.000128	0.000202	0.000254	0.000068	0.000077	0.000052	0	0.000416	0	0	-3.210461	2.865366	2.876749	2
0.004164	151.678875	0.000096	4.800396	0	0	0.00012	0.00019	0.000238	0.000064	0.000072	0.000048	0	0.000363	0	0	-3.276674	2.910968	2.905399	2
0.006368	232.445949	0.000223	11.087198	0	0	0.000312	0.000439	0.000614	0.000131	0.000169	0.000109	0	0.002452	0	0	-2.327634	2.803804	2.806195	2
0.005155	187.93768	0.000144	7.177375	0	0	0.000189	0.000284	0.000372	0.000091	0.000108	0.000069	0	0.000902	0	0	-2.827104	2.810916	2.828245	2
0.006788	677.934999	0.000256	34.801056	7	10	0.000436	0.000502	0.000853	0.000131	0.000189	0.000095	0	0.013192	0	7	-1.49275	2.735154	2.738337	2

0.008762	878.07212	0.000417	56.957604	83	114	0.000752	0.000821	0.001479	0.000188	0.000305	0.000167	0	0.039253	0.000001	83	-0.942442	2.798012	2.788207	2
0.008827	896.289845	0.00053	72.310835	47	66	0.000866	0.001042	0.001703	0.000232	0.000425	0.000301	0	0.052083	0.000001	47	-0.801499	2.791409	2.783324	2
0.008937	911.63818	0.000651	88.635112	534	722	0.001225	0.001278	0.002402	0.000207	0.000534	0.00038	0	0.104099	0.000002	534	-0.457949	2.761637	2.754324	2
0.004004	82.040566	0.000091	2.569164	0	0	0.000115	0.000181	0.000227	0.000061	0.000069	0.000045	0	0.000188	0	0	-3.608012	2.889507	2.883074	2
0.005223	107.058258	0.000156	4.377343	0	0	0.000245	0.000308	0.000484	0.000091	0.000113	0.000067	0	0.000855	0	0	-2.853507	2.818656	2.821399	2
0.007087	144.825696	0.000262	7.318902	0	0	0.000425	0.000515	0.000834	0.000136	0.000187	0.00015	0	0.002568	0	0	-2.308046	2.774403	2.768166	2
0.005138	105.914374	0.000165	4.638688	0	0	0.000231	0.000327	0.000455	0.000101	0.000124	0.000082	0	0.000755	0	0	-2.915485	2.813436	2.82791	2
0.004238	91.171498	0.0001	2.961305	0	0	0.000129	0.000199	0.000255	0.000065	0.000076	0.000052	0	0.000248	0	0	-3.468518	2.860317	2.874873	2
0.004109	88.403861	0.000095	2.81062	0	0	0.00012	0.000188	0.000237	0.000063	0.000071	0.000049	0	0.000213	0	0	-3.54303	2.913732	2.908232	2
0.005347	115.68947	0.000186	5.468402	0	0	0.000271	0.000367	0.000532	0.000108	0.000144	0.000119	0	0.001092	0	0	-2.73441	2.783732	2.789449	2
0.004464	96.242361	0.000116	3.415252	0	0	0.000146	0.000229	0.000288	0.000076	0.000088	0.00006	0	0.000317	0	0	-3.346199	2.876433	2.88547	2
0.007921	213.581965	0.000335	12.306934	6	9	0.000584	0.000658	0.001147	0.000153	0.000253	0.000149	0	0.006377	0	6	-1.85211	2.788456	2.777769	2
0.004717	127.178493	0.000119	4.396084	0	0	0.000151	0.000235	0.000298	0.000078	0.000089	0.000058	0	0.000424	0	0	-3.201588	2.848745	2.862009	2
0.007411	199.627865	0.00029	10.67591	0	0	0.000484	0.000571	0.000949	0.00014	0.00022	0.000131	0	0.004376	0	0	-2.041532	2.77076	2.767398	2
0.007257	195.799151	0.000287	10.571182	0	0	0.000475	0.000565	0.000934	0.00014	0.000218	0.000125	0	0.004217	0	0	-2.057886	2.792653	2.787374	2
0.006177	176.751018	0.000247	9.622333	0	0	0.000454	0.000487	0.000894	0.000118	0.000194	0.00015	0	0.004065	0	0	-2.074735	2.810949	2.805139	2
0.004351	123.979673	0.000108	4.213955	0	0	0.000136	0.000213	0.00027	0.000072	0.000082	0.000055	0	0.000367	0	0	-3.272555	2.853573	2.869155	2
0.006636	189.98825	0.000271	10.518251	0	0	0.000407	0.000533	0.0008	0.000145	0.00021	0.000154	0	0.00327	0	0	-2.185508	2.791692	2.78512	2
0.00495	140.726215	0.000142	5.50465	0	0	0.000215	0.000279	0.000423	0.000083	0.000107	0.000078	0	0.000915	0	0	-2.822507	2.784562	2.787812	2
0.00652	238.383225	0.000265	13.186244	0	0	0.000462	0.000521	0.000907	0.000119	0.000198	0.000145	0	0.005405	0	0	-1.936429	2.769484	2.7612	2
0.005034	183.670621	0.00014	6.987275	0	0	0.000214	0.000276	0.000422	0.000083	0.000103	0.000061	0	0.001159	0	0	-2.701281	2.809385	2.821129	2
0.00643	234.751521	0.000242	12.047198	0	0	0.000407	0.000476	0.000801	0.000119	0.000179	0.000131	0	0.004198	0	0	-2.060911	2.782986	2.781532	2
0.00713	260.281159	0.000303	15.062905	1	2	0.00057	0.000595	0.00112	0.000129	0.000222	0.000153	0	0.008231	0	1	-1.725147	2.776115	2.771859	2
0.02669	25.126965	0.003515	4.455646	94	104	0.008134	0.006973	0.016136	0.001436	0.002545	0.002245	0.035994	0.042275	0.000066	94	-0.897511	2.905863	2.900756	1
0.024757	733.849871	0.00312	123.850823	2984	3387	0.006566	0.006193	0.013025	0.001298	0.002358	0.001467	0.0304	0.862205	0.000043	2986	0.610859	2.909478	2.908311	1
0.023475	693.503959	0.00264	104.703607	2735	3140	0.00488	0.005235	0.009673	0.001237	0.001996	0.001367	0.01695	0.476288	0.000024	2735	0.313317	2.893354	2.898417	1

Tabla 20. Clasificador 3, con datos de entrada en la red neuronal representando a las columnas un total de 52 sujetos de entrenamiento (49 enfermos y 3 sanos)

0.035957	1066.95677	0.005645	224.020289	6412	7128	0.010446	0.011201	0.020726	0.002263	0.004146	0.002595	0.07995	2.182185	1.09E-04	6414	1.075348	2.913495	2.917005	1
0.02669	25.126965	3.52E-03	4.455646	94	104	0.008134	0.006973	0.016136	1.44E-03	2.55E-03	2.25E-03	0.035994	0.042275	6.60E-05	94	0.897511	2.905863	2.900756	1
0.024757	733.849871	3.12E-03	123.850823	2984	3387	0.006566	0.006193	0.013025	1.30E-03	2.36E-03	1.47E-03	0.0304	0.862205	4.30E-05	2986	0.610859	2.909478	2.908311	1
0.023475	693.503959	0.00264	104.703607	2735	3140	0.00488	0.005235	0.009673	1.24E-03	0.001996	1.37E-03	0.01695	0.476288	2.40E-05	2735	0.313317	2.893354	2.898417	1
0.004165	91.80227	9.40E-05	2.856835	0	0	0.000118	0.000187	0.000233	6.30E-05	7.10E-05	4.80E-05	0	0.000212	0	0	3.546894	2.87877	2.88319	1
0.003856	84.989574	8.30E-05	2.517872	0	0	0.000104	0.000165	0.000206	5.50E-05	6.20E-05	4.10E-05	0	0.000165	0	0	3.671084	2.899708	2.896554	1
0.005103	112.696766	1.45E-04	4.366873	0	0	0.000193	0.000285	0.00038	9.10E-05	1.09E-04	7.20E-05	0	0.00057	0	0	3.056865	2.807612	2.81536	1
0.005058	111.404244	0.000138	4.161699	0	0	0.000181	0.000272	0.000356	8.90E-05	0.000102	6.50E-05	0	0.000499	0	0	3.123864	2.803426	2.814084	1
0.004165	91.80227	9.40E-05	2.856835	0	0	0.000118	0.000187	0.000233	6.30E-05	7.10E-05	4.80E-05	0	0.000212	0	0	3.546894	2.87877	2.88319	1
0.003856	84.989574	8.30E-05	2.517872	0	0	0.000104	0.000165	0.000206	5.50E-05	6.20E-05	4.10E-05	0	0.000165	0	0	3.671084	2.899708	2.896554	1
0.005103	112.696766	0.000145	4.366873	0	0	0.000193	0.000285	0.00038	9.10E-05	0.000109	7.20E-05	0	0.00057	0	0	3.056865	2.807612	2.81536	1
0.005058	111.404244	0.000138	4.161699	0	0	0.000181	0.000272	0.000356	8.90E-05	0.000102	6.50E-05	0	0.000499	0	0	3.123864	2.803426	2.814084	1
0.004352	128.575469	0.000104	4.218346	0	0	0.000131	0.000206	0.000258	6.90E-05	7.80E-05	5.20E-05	0	0.000349	0	0	3.297122	2.863336	2.877988	1
0.005099	150.568926	0.000142	5.730833	0	0	0.000187	0.00028	0.000369	9.20E-05	0.000107	7.30E-05	0	0.000719	0.00E+00	0	2.940833	2.799948	2.815815	1
0.005626	166.480771	0.000184	7.42425	0	0	0.000253	0.000362	0.000497	0.000112	0.000142	0.000102	0	0.001314	0.00E+00	0	2.642288	2.767987	2.783213	1
0.00464	136.847886	1.23E-04	4.954725	0	0	1.62E-04	2.42E-04	3.19E-04	7.80E-05	9.30E-05	6.30E-05	0	0.000541	0	0	3.086957	2.76E+00	2.785548	1
0.012876	5126.3826	7.85E-04	424.31723	210	260	1.05E-03	0.001556	2.08E-03	4.74E-04	5.93E-04	4.10E-04	0	0.298231	1.00E-06	210	0.081642	2.95E+00	2.944427	1
0.01248	4598.94237	0.000758	380.964864	185	232	0.001026	0.001501	0.002038	0.000437	0.00057	0.000386	0.00E+00	0.267249	1.00E-06	185	0.026589	2.945929	2.942322	1
0.003171	237.543052	5.00E-05	2.445363	0	0	5.20E-05	1.50E-05	2.60E-05	4.90E-05	3.80E-05	2.50E-05	0	0.000437	0	0	4.567574	1.00E-05	0.183356	1
0.003203	22.597608	5.10E-05	0.04587	0	0	5.10E-05	0	1.60E-05	5.00E-05	3.80E-05	2.50E-05	0	0.000428	0	0	5.030964	1.00E-05	0.073858	1
0.017213	1923.61986	0.001566	235.058578	4366	4778	0.002537	0.003116	0.005054	0.000828	0.001234	0.000807	9.30E-05	0.485492	6.00E-06	4366	0.327935	2.999236	2.990959	1
0.035957	1066.95677	0.005645	224.020289	6412	7128	0.010446	0.011201	0.020726	0.002263	0.004146	0.002595	0.07995	2.182185	0.000109	6414	1.075348	2.913495	2.917005	2
0.038145	782.291008	0.006497	177.329868	4832	5358	0.012128	0.012894	0.02406	0.002619	0.004932	0.003423	0.099033	2.022765	0.000147	4837	1.037247	2.905755	2.913555	2
0.036898	223.617561	0.007154	57.510854	1035	1101	0.016921	0.014186	0.033523	0.002094	0.005406	0.003816	0.125802	1.160786	2.86E-04	1052	0.758083	2.875839	2.883835	2
0.03843	1143.9463	0.00756	299.63037	5091	5426	0.021317	0.014982	0.042227	0.002238	0.005605	0.003738	0.12645	9.088375	4.54E-04	5136	1.787034	2.878989	2.884153	2
0.03843	1143.9463	0.00756	299.63037	5091	5426	0.021317	0.014982	0.042227	2.24E-03	0.005605	3.74E-03	0.12645	9.088375	0.000454	5136	1.787034	2.878989	2.884153	2

0.013097	257.301304	0.000828	22.174368	226	291	0.001553	0.001621	0.003039	2.50E-04	5.29E-04	1.54E-04	0	0.033	2.00E-06	226	1.034654	2.744238	2.731782	2
0.018394	366.609073	0.001737	46.726303	1482	1827	0.00303	0.003416	0.005958	0.000454	0.001215	0.000651	0.000731	0.125605	9.00E-06	1482	0.361329	2.788802	2.781836	2
0.005435	107.261047	0.000161	4.346737	0	0	0.00023	0.000318	0.000455	9.90E-05	0.000121	7.70E-05	0	0.000726	0.00E+00	0	-2.9344	2.81806	2.830905	2
0.004709	93.029797	0.000122	3.297407	0	0	0.000154	0.000241	0.000305	8.00E-05	9.10E-05	6.00E-05	0	0.000326	0	0	3.333994	2.847234	2.862166	2
0.013097	257.301304	0.000828	22.174368	226	291	0.001553	0.001621	0.003039	2.50E-04	5.29E-04	1.54E-04	0	0.033	2.00E-06	226	1.034654	2.744238	2.731782	2
0.018394	366.609073	1.74E-03	46.726303	1482	1827	0.00303	0.003416	0.005958	4.54E-04	1.22E-03	6.51E-04	0.000731	0.125605	9.00E-06	1482	0.361329	2.788802	2.781836	2
0.005435	107.261047	1.61E-04	4.346737	0	0	0.00023	0.000318	0.000455	9.90E-05	1.21E-04	7.70E-05	0	0.000726	0	0	-2.9344	2.81806	2.830905	2
0.004709	93.029797	0.000122	3.297407	0	0	0.000154	0.000241	0.000305	8.00E-05	9.10E-05	6.00E-05	0	0.000326	0	0	3.333994	2.847234	2.862166	2
0.00419	235.659418	9.90E-05	7.631819	0	0	0.000124	0.000196	0.000246	6.50E-05	7.40E-05	5.00E-05	0	0.0006	0	0	3.025914	2.87714	2.888217	2
0.004082	229.671767	9.40E-05	7.265767	0	0	0.000118	0.000186	0.000234	6.20E-05	7.10E-05	4.80E-05	0	0.000541	0	0	3.076798	2.92045	2.91362	2
0.004994	280.248371	0.000131	10.080105	0	0	0.000167	0.000259	0.00033	8.60E-05	9.80E-05	6.70E-05	0	0.001092	0.00E+00	0	-2.73032	2.824438	2.842342	2
0.004651	261.387007	0.000117	9.011757	0	0	0.000147	0.000231	0.000291	7.70E-05	8.80E-05	6.00E-05	0	0.000844	0	0	2.855912	2.870249	2.881572	2
0.005143	223.75364	0.000158	9.344021	0	0	0.000214	0.000311	0.000422	9.80E-05	0.000122	8.00E-05	0	0.00138	0	0	2.614797	2.80748	2.815017	2
0.006362	280.504826	0.000353	20.826783	66	93	0.000767	0.000693	0.0015	0.000126	0.000303	0.00025	0	0.017711	1.00E-06	66	1.346382	2.732656	2.72937	2
0.006979	307.721508	0.000361	21.374902	18	27	0.000622	0.000711	0.001224	0.000176	0.000296	0.000205	0	0.011653	0	18	1.550102	2.787603	2.783901	2
0.006661	290.722506	0.000293	17.304906	28	37	0.000559	0.000575	0.001097	0.000134	0.000225	0.000121	0	0.009397	0	28	1.659466	2.776496	2.766498	2

Tabla 21. Clasificador 4, con datos de entrada en la red neuronal representando a las columnas un total de 42 sujetos para entrenamiento (21 pacientes sanos y 21 pacientes enfermos)

0.02669	25.126965	0.003515	4.455646	94	104	0.008134	0.006973	0.016136	0.001436	0.002545	0.002245	0.035994	0.042275	0.000066	94	-0.897511	2.905863	2.900756	1
0.024757	733.849871	0.00312	123.850823	2984	3387	0.006566	0.006193	0.013025	0.001298	0.002358	0.001467	0.0304	0.862205	0.000043	2986	0.610859	2.909478	2.908311	1
0.023475	693.503959	0.00264	104.703607	2735	3140	0.00488	0.005235	0.009673	0.001237	0.001996	0.001367	0.01695	0.476288	0.000024	2735	0.313317	2.893354	2.898417	1
0.036608	2682.36851	0.006064	590.969572	16749	18583	0.011031	0.012032	0.021883	0.002491	0.004598	0.00314	0.089153	5.976443	0.000122	16759	1.57893	2.908455	2.914274	2
0.037054	1102.48561	0.006061	240.485786	6874	7624	0.010839	0.012024	0.021503	0.002503	0.004542	0.002879	0.09135	2.349555	0.000117	6877	1.112169	2.908901	2.913703	2
0.036718	1093.73183	0.006112	242.577121	6691	7433	0.011403	0.012129	0.022619	0.00244	0.004537	0.002964	0.0897	2.600543	0.00013	6696	1.162748	2.905824	2.91243	2

Tabla 22. Datos para pruebas en la red neuronal representando a las columnas un total de 6 sujetos para entrenamiento (3 pacientes sanos y 3 pacientes enfermos)

0.000062	0.000041	0	0.000165	0	0	-3.671084	2.899708	2.896554	1
0.000109	0.000072	0	0.00057	0	0	-3.056865	2.807612	2.81536	1
0.000102	0.000065	0	0.000499	0	0	-3.123864	2.803426	2.814084	1
0.000071	0.000048	0	0.000212	0	0	-3.546894	2.87877	2.88319	1
0.000062	0.000041	0	0.000165	0	0	-3.671084	2.899708	2.896554	1
0.000109	0.000072	0	0.00057	0	0	-3.056865	2.807612	2.81536	1
0.000102	0.000065	0	0.000499	0	0	-3.123864	2.803426	2.814084	1
0.000078	0.000052	0	0.000349	0	0	-3.297122	2.863336	2.877988	1
0.000107	0.000073	0	0.000719	0	0	-2.940833	2.799948	2.815815	1
0.000142	0.000102	0	0.001314	0	0	-2.642288	2.767987	2.783213	1
0.000093	0.000063	0	0.000541	0	0	-3.086957	2.759676	2.785548	1
0.004598	0.00314	0.089153	5.976443	0.000122	16759	1.57893	2.908455	2.914274	2
0.004542	0.002879	0.09135	2.349555	0.000117	6877	1.112169	2.908901	2.913703	2
0.004537	0.002964	0.0897	2.600543	0.00013	6696	1.162748	2.905824	2.91243	2
0.00482	0.003151	0.091055	3.194605	0.00016	7087	1.266341	2.921664	2.924728	2
0.000121	0.000077	0	0.000726	0	0	-2.9344	2.81806	2.830905	2
0.005528	0.003551	0.12795	14.016098	0.00035	10173	2.003667	2.877658	2.884285	2
0.005688	0.003631	0.13355	7.798316	0.00039	5234	1.710326	2.875117	2.880638	2
0.005344	0.003493	0.12235	6.217787	0.000311	4939	1.597511	2.878465	2.888861	2
0.000103	0.000061	0	0.001159	0	0	-2.701281	2.809385	2.821129	2
0.000179	0.000131	0	0.004198	0	0	-2.060911	2.782986	2.781532	2
0.000222	0.000153	0	0.008231	0	1	-1.725147	2.776115	2.771859	2

Tabla 23. Clasificador 5, con datos de entrada en la red neuronal representando a las columnas un total de 22 sujetos para entrenamiento (11 pacientes sanos y 11 pacientes enfermos)

4.5 Resultados

Se hicieron pruebas para 79 muestras (Tabla 18), 74 muestras (Tabla 19), 52 muestras (Tabla 20), 42 muestras (Tabla 21), 22 muestras (Tabla 23) con el objetivo de encontrar la que mejor se ajusta al problema que se tiene, por ello se probaron con diversas muestras en diferentes proporciones.

La Tabla 21 se encuentra balanceada, ya que se tienen 21 pacientes sanos y 21 pacientes enfermos, se prueba con datos de la Tabla 22, más adelante se detallan los resultados.

La Tabla 23 se encuentra balanceada completamente porque se tienen 11 pacientes sanos y 11 pacientes enfermos, además se hizo una tabla con 6 muestras para pruebas mostradas en la Tabla 22 y probar su efectividad, más adelante se detallan los resultados.

Se ingresaron los datos de entrenamiento a la red neuronal en Matlab, al igual que los datos de prueba. Así como sus vectores objetivo. Debido a la cantidad de pruebas realizadas solo se muestran las más significativas.

La Figura 27, Figura 28, Figura 29 y Figura 30 muestran las diferentes gráficas de salida ((a) rendimiento, (b) estado de entrenamiento, (c) gráfica de regresión) para cada una de las pruebas descritas en el pie de figura. Las pruebas fueron realizadas con los datos de las tablas previamente descritas. Los resultados de esta etapa son bajos, por lo que es importante destacar que, debido a ello, se crearon los distintos datos de entrada o entrenamiento para la mejora del modelo.

El valor del gradiente se visualiza en la Figura (a), que expresa el grado de ajuste entre los datos de entrenamiento y prueba contra los datos de salida reales. El error cuadrático medio (por sus siglas en inglés MSE), es la diferencia cuadrática media entre salidas (outputs) y targets (objetivos). Los valores cercanos a 0 o iguales significa que no hubo errores.

Si el modelo es bueno, entonces el MSE de datos de prueba (test) es similar al conjunto de datos de entrenamiento (train). De lo contrario se cumplen dos condiciones:

- Resultado MSE de prueba > Resultado MSE de train = Sobre ajuste de los datos.
- Resultado MSE de prueba < Resultado MSE de train = Bajo ajuste de los datos.

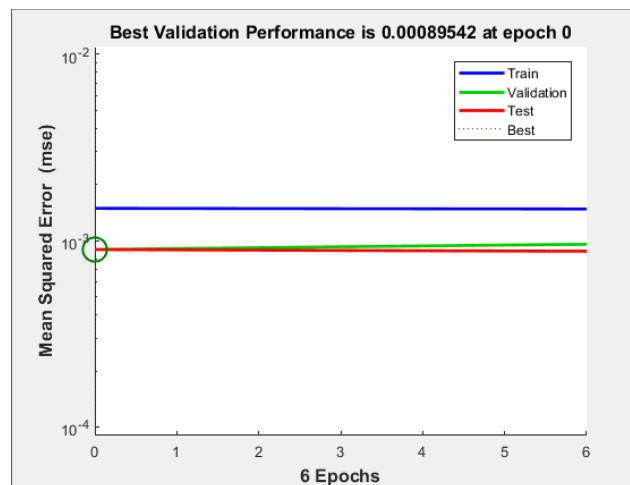
La Figura (b) muestra el estado de la red durante el proceso de aprendizaje. Se puede ver el cambio de gradiente específico del algoritmo BP con el aumento de iteraciones directamente. Si el valor del error aumenta, entonces el valor lr (learning rate) aumentará en consecuencia, por lo que cuando el valor de lr es demasiado grande, el proceso de aprendizaje debe detenerse.

La Figura (c) muestra los resultados de la red con respecto a los objetivos para los conjuntos de entrenamiento, validación y prueba. Para un resultado perfecto, los cuatro gráficos deben tener una línea de 45 grados; dónde las salidas de la red son iguales a los objetivos. Los resultados correspondientes indican la salida deseada contra la salida real, 1 significa mucha relación, mientras que 0 significa poca o nula relación.

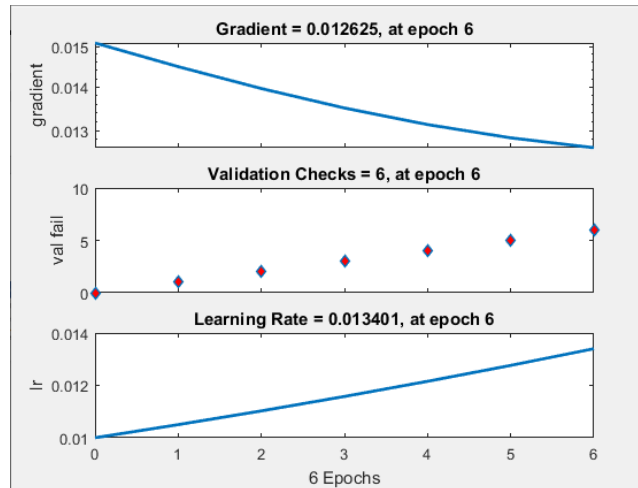
4.5.1 Resultados bajos

4.5.1.1 Resultado prueba 86 con 52 muestras

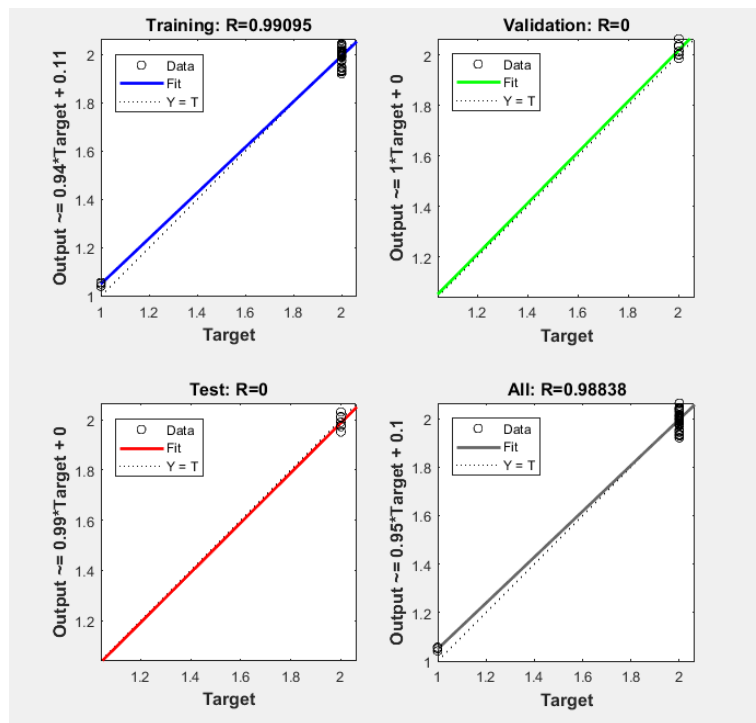
Las gráficas de la Figura 27 de la prueba 86 muestran poca precisión (a) y aunque el ajuste en la gráfica (c) es regular en 2 de 4 parámetros, no se logra el objetivo de tener la mayor precisión posible, lo que se puede observar en la gráfica (b) que el aprendizaje disminuye mientras sigue con el entrenamiento, lo que indica falta de datos para seguir aprendiendo.



(a) Datos predictivos y reales. Este entrenamiento se detuvo cuando el error de validación aumentó durante seis iteraciones.



(b) Estado de la red. El gradiente disminuye, y no hay más iteraciones, por lo que el modelo no es válido para este problema.

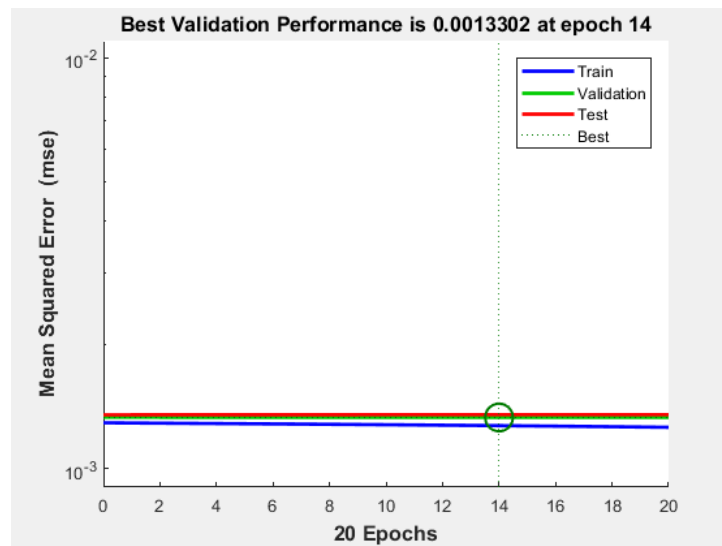


(c) Gráfica de regresión: Para esta configuración, el ajuste no es bueno, en todos los conjuntos de datos debería tener valores de R superior a 0.9 o más.

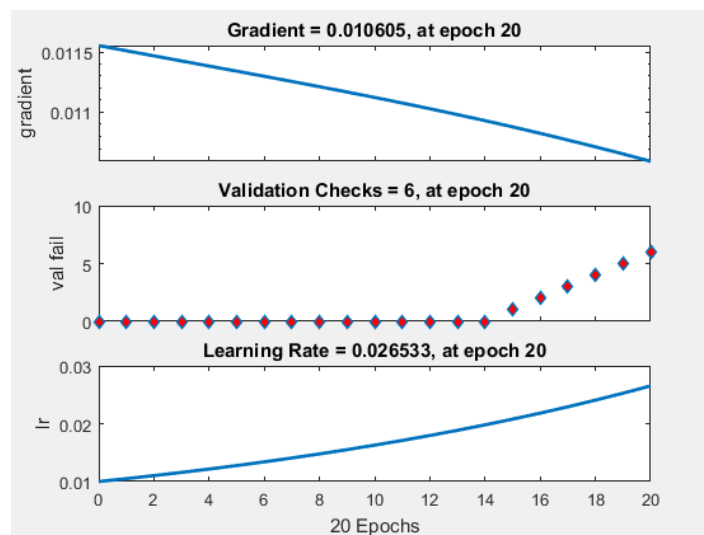
Figura 27. Resultado del experimento número 86

4.5.1.2 Resultado prueba 87 con 52 muestras

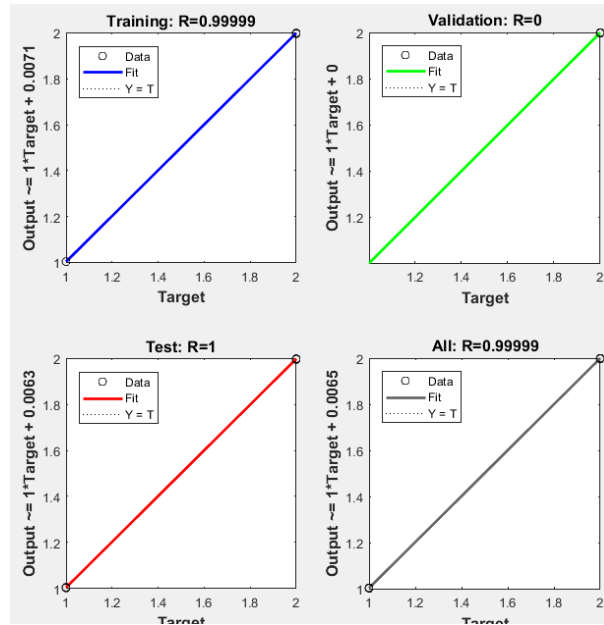
Las gráficas de la Figura 28 de la prueba 87 muestran bajo rendimiento (a) y aunque el ajuste en la gráfica (c) es bueno en 3 de 4 parámetros, no se logra el objetivo de tener la mayor precisión posible, lo que se puede observar en la gráfica (b) que el aprendizaje llega a su punto máximo en el entrenamiento en el período 20, lo que puede indicar falta de datos para aprender.



(a) Datos Predictivos y reales. Este entrenamiento se detuvo cuando el error de validación aumentó durante 20 iteraciones.



(b) Estado de la red. El gradiente disminuye, y no hay más iteraciones, y el valor de learning rate es bueno.

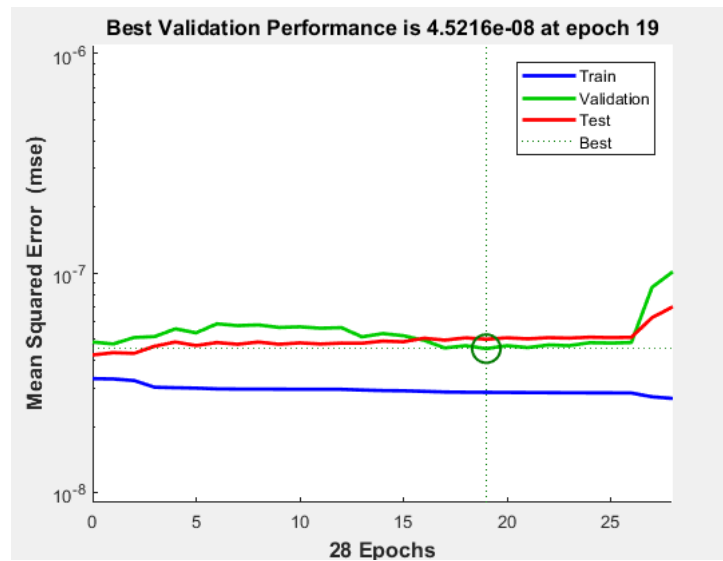


(c) Gráfica de regresión: Para esta configuración, el ajuste es regular, en todos los conjuntos de datos debería tener valores de R superior a 0.9 o más.

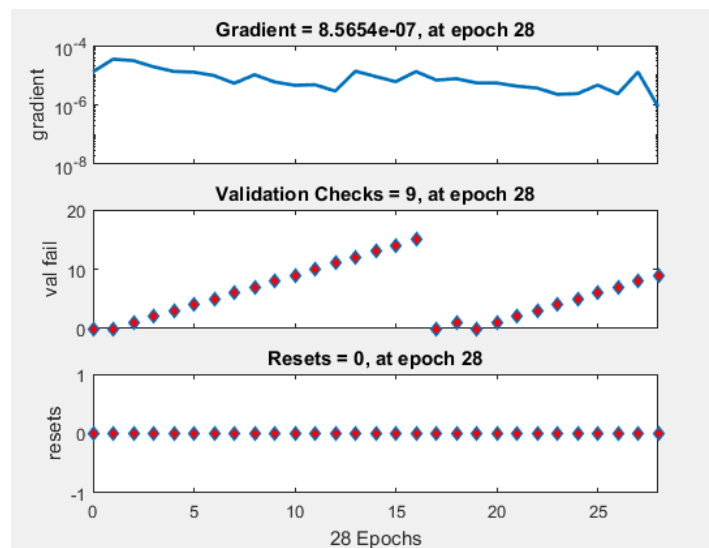
Figura 28. Resultados del experimento número 87

Las gráficas de la Figura 29 de la prueba 89 muestran rendimiento aceptable (a) y aunque el ajuste en la gráfica (c) es bueno en 3 de 4 parámetros, no se logra el objetivo de tener la mayor precisión posible, lo que se puede observar en la gráfica (b) que el aprendizaje llega a su punto máximo en el período 1, lo que indica falta de datos para aprender.

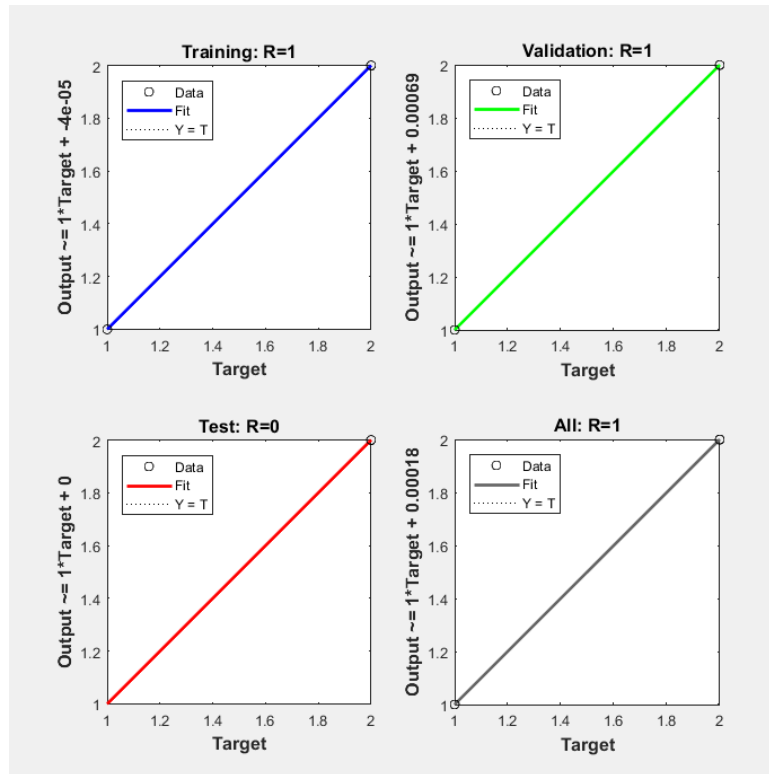
4.5.1.3 Resultado prueba 89 con 52 muestras



(a) Datos predictivos y reales. Este entrenamiento se detuvo cuando el error de validación aumentó durante 28 iteraciones.



(b) Estado de la red. El gradiente sube y baja, se detiene en iteración 28. Los valores de error tienen una subida constante.

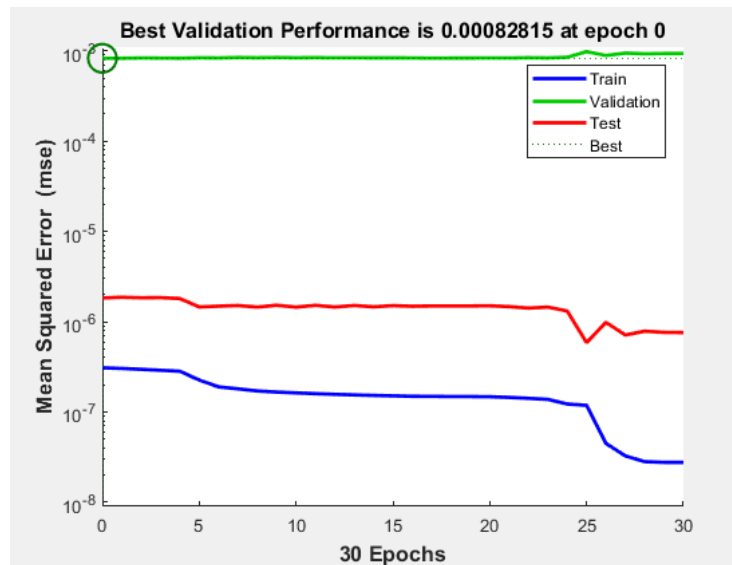


(c) Gráfica de regresión: Para esta configuración, el ajuste es bajo, en todos los conjuntos de datos debería tener valores de R superior a 0.9 o más.

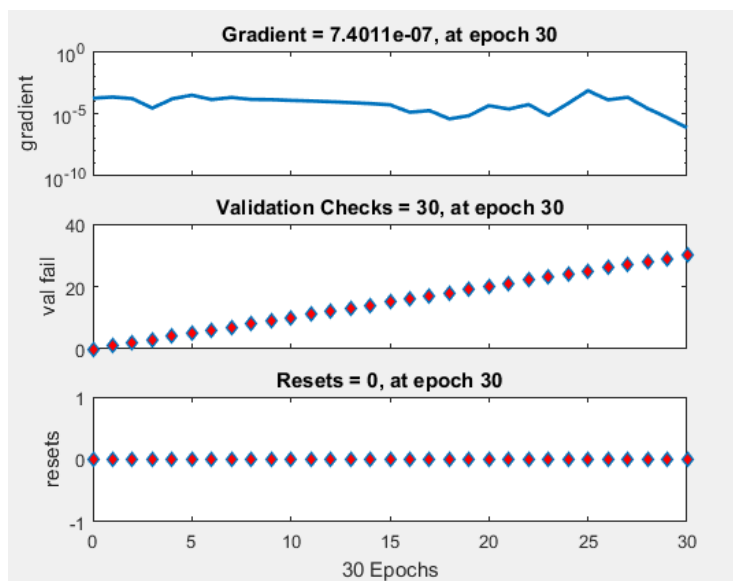
Figura 29. Resultado del experimento número 89

Las gráficas de la Figura 30 de la prueba 90 muestran rendimiento bajo pues muestra un bajo ajuste de los datos (a) y aunque el ajuste en la gráfica (c) es bueno en 3 de 4 parámetros, no se logra el objetivo de tener la mayor precisión posible, lo que se puede observar en la gráfica (b) que el aprendizaje llega a su punto máximo en el entrenamiento en el período 1.

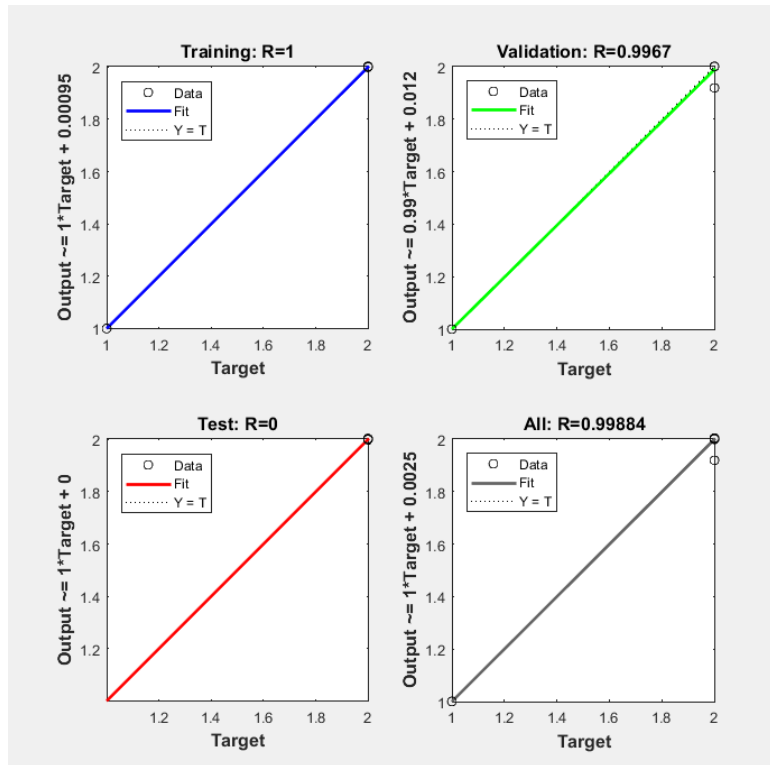
4.5.1.4 Resultado prueba 90 con 52 muestras



- (a) Datos Predictivos y reales. Este entrenamiento se detuvo cuando el error de validación aumentó durante 30 iteraciones. Se observa sobre ajuste en los datos.



- (b) Estado de la red. El gradiente disminuye, y el error aumenta.



- (c) Gráfica de regresión: Para esta configuración, el ajuste no es bueno, en todos los conjuntos de datos debería tener valores de R superior a 0.9 o más.

Figura 30. Resultados del experimento número 90

La Figura 27, Figura 28, Figura 29 y Figura 30 son pruebas realizadas y mostradas para comprobar la funcionalidad de las pruebas con las configuraciones dadas en la Tabla 17 sin llegar a ser eficientes. A continuación, se muestran los diferentes resultados que alcanzan el objetivo de ajuste de los datos.

4.5.2 Resultados con mejor rendimiento

4.5.2.1 Resultado prueba 108 con 52 muestras

La Figura 31 muestra la gráfica de desempeño del experimento 108, se observa sobre ajuste de los datos, esa prueba se hizo con los datos de la Tabla 19, los cuales son 21 sanos y 53 enfermos, se nota el efecto que tiene el sobre ajuste de los datos.

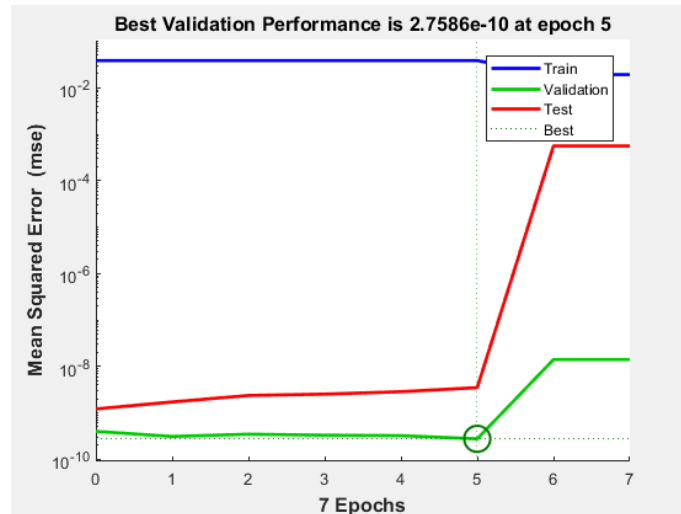


Figura 31. Gráfica de desempeño del experimento número 108

Este entrenamiento se detuvo en la sexta iteración, el mejor rendimiento de validación es 0.000000000027586 (MSE). Un error bajo, lo que indica estar cercano al resultado, pero al tener más datos enfermos que sanos, puede que lo aprendido no sea real.

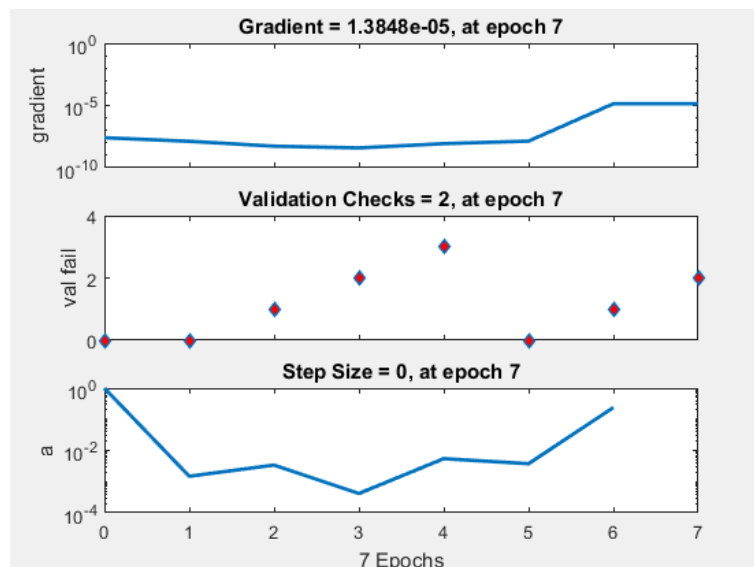


Figura 32. Gráfica del estado de entrenamiento del experimento número 108

El gradiente en la Figura 32 expresa el grado de ajuste entre los datos de entrenamiento y prueba y los datos de salida reales.

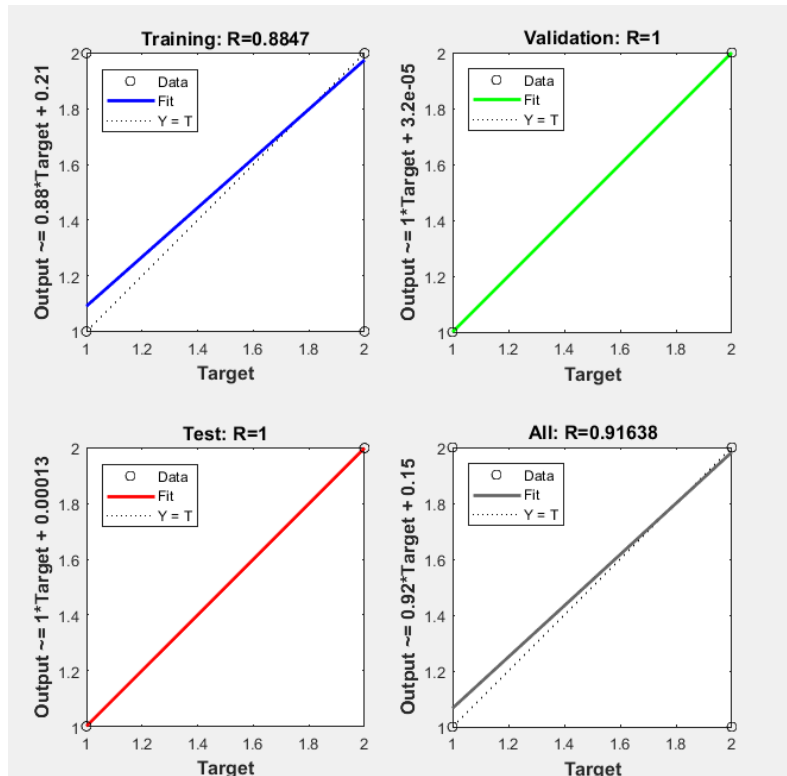


Figura 33. Gráfica de regresión del experimento número 108

Como se explica anteriormente, los valores R de la gráfica de regresión de la Figura 33, es la correlación entre las salidas y los objetivos. Un valor R de 1 significa una relación cercana, un valor 0 una relación aleatoria. Se entiende que los valores biomédicos son aleatorios, no son iguales, a pesar de esto, la relación es cercana a la predicción de estos en la red neuronal.

En el anexo 4, se muestran diferentes pruebas con la combinación de los datos de la Tabla 23 con los resultados más significativos. Tiene resultados satisfactorios, pero como se observan los datos de la tabla no están balanceados y los resultados son poco confiables.

4.5.2.2 Resultado con 42 muestras

Se hicieron experimentos con 42 muestras (21 sujetos enfermos y 21 sujetos sanos) que se pueden observar en la Tabla 21, de los cuales se tomaron aleatoriamente 30 muestras para entrenamiento (70%), 8 muestras para validación (20%) y 4 muestras para pruebas (10%), con 10 neuronas ocultas. El algoritmo de entrenamiento con aprendizaje no supervisado es trainlm. El resultado de la gráfica de rendimiento es:

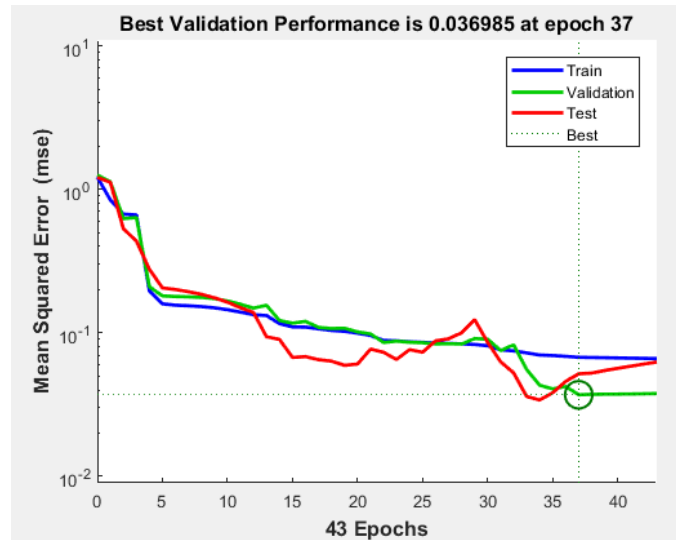


Figura 34. Gráfica de evaluación del desempeño de la Tabla 21.

En la Figura 34 se visualiza el rendimiento muy bueno, ya que está cercano al 0.

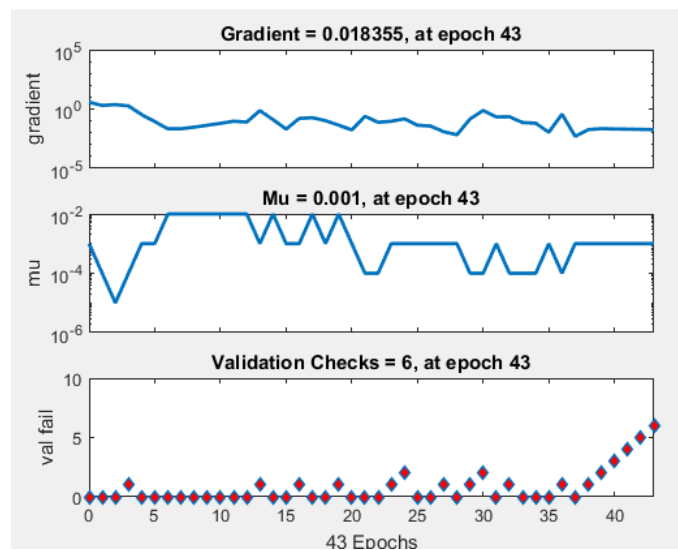


Figura 35. Gráfica del estado de entrenamiento de la Tabla 21

En la Figura 35 se visualiza el grado de ajuste entre los datos de aprendizaje y los de prueba, alcanza un valor de fallo mínimo.

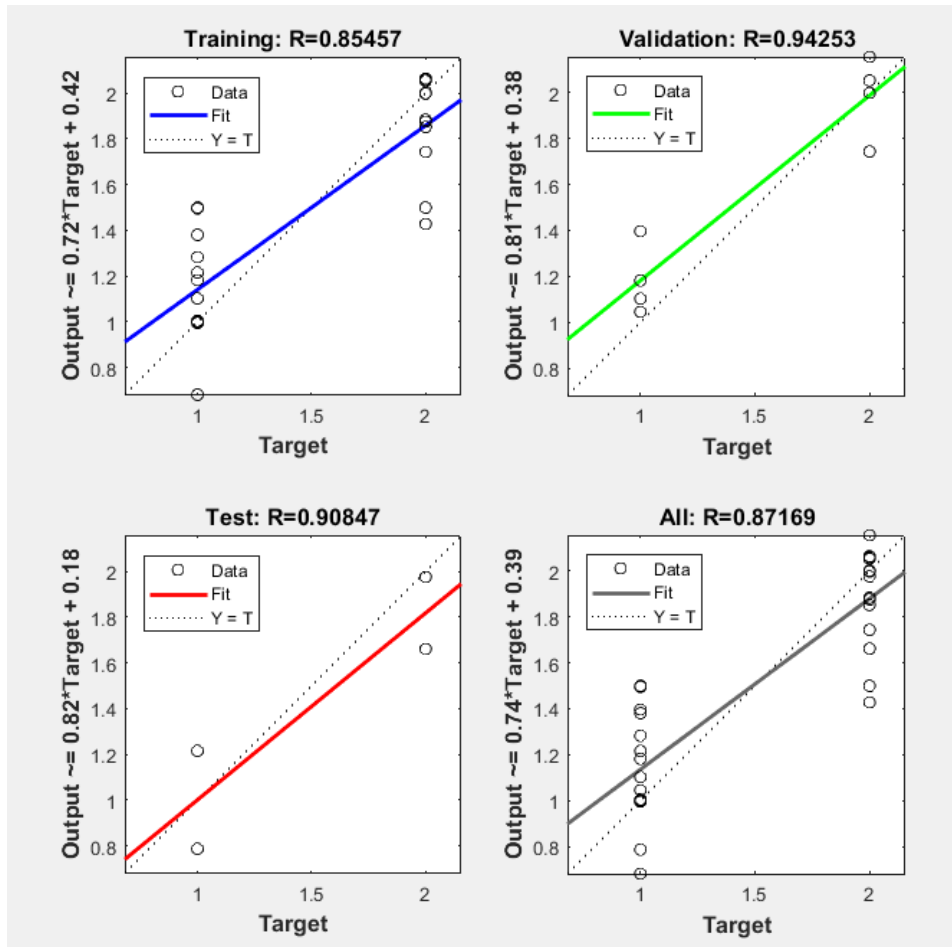


Figura 36. Gráfica de regresión del entrenamiento de la Tabla 21

La Figura 36 muestra cercanía con el ángulo de 45° y cercanos a la unidad, el resultado es de 89% de eficacia del modelo. Lo que lo hace cercano a nuestro objetivo que sea la efectividad mayor al 90%.

4.5.2.3 Resultado con 74 muestras

Se hicieron experimentos con 74 muestras (59 sujetos enfermos y 15 sujetos sanos) que se pueden observar en la Tabla 19, de los cuales se tomaron aleatoriamente 48 muestras para entrenamiento (70%), 15 muestras para validación (20%) y 11 muestras para pruebas (10%), con 10 neuronas ocultas. El algoritmo de entrenamiento con aprendizaje no supervisado es trainlm. El resultado de la gráfica de rendimiento es:

Train Network
Train the network to fit the inputs and targets.

Train Network

Choose a training algorithm:

Levenberg-Marquardt

This algorithm typically requires more memory but less time. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Train using Levenberg-Marquardt. (trainlm)

Retrain

Results

	Samples	MSE	R
Training:	48	1.27586e-2	9.53058e-1
Validation:	15	1.88086e-2	9.40568e-1
Testing:	11	1.83946e-1	8.01760e-1

Plot Fit Plot Error Histogram

Plot Regression

Notes

Training multiple times will generate different results due to different initial conditions and sampling.

Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.

Regression R Values measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship.

Figura 37. Resultados del entrenamiento de la Tabla 19

La Figura 37 muestra el resultado con un rendimiento del 89%, para 74 muestras, lo que demuestra una relación cercana.

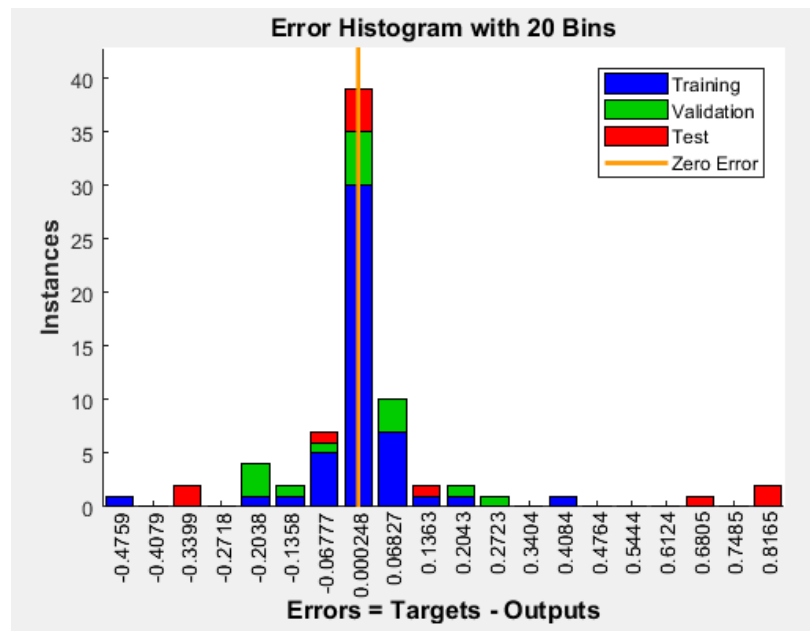


Figura 38. Histograma de error con 74 muestras

La Figura 38 muestra el histograma de error de las 74 muestras, y realmente casi no hubo errores pues en el punto 0.000248 se encuentran las tres instancias (entrenamiento, validación y pruebas).

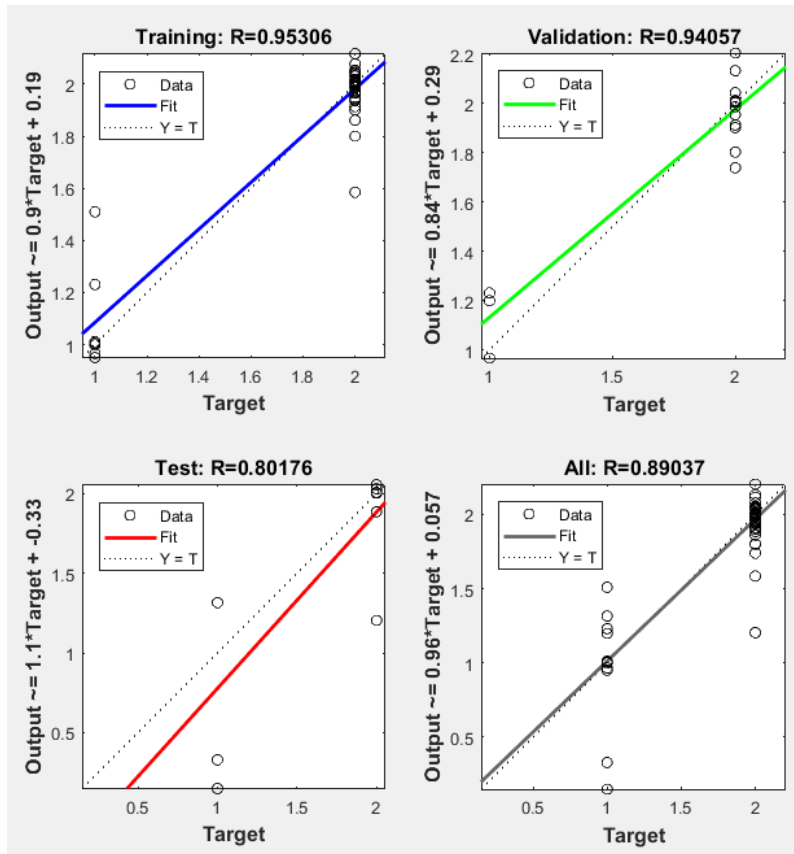


Figura 39. Gráfica de regresión del entrenamiento de la Tabla 19

La Figura 39 muestra cercanía con el ángulo de 45° y cercanos a la unidad, el resultado es de 89% de eficacia del modelo. Lo que lo hace cercano a nuestro objetivo que sea la efectividad mayor al 90%.

4.5.2.4 Resultado con 79 muestras

Con las muestras de la Tabla 18, se tomaron aleatoriamente 55 muestras para entrenamiento (70%), 16 muestras para validación (20%) y 8 muestras para pruebas (10%), con 10 neuronas ocultas. El algoritmo de entrenamiento con función de transferencia trainlm. El resultado de la gráfica de rendimiento es:

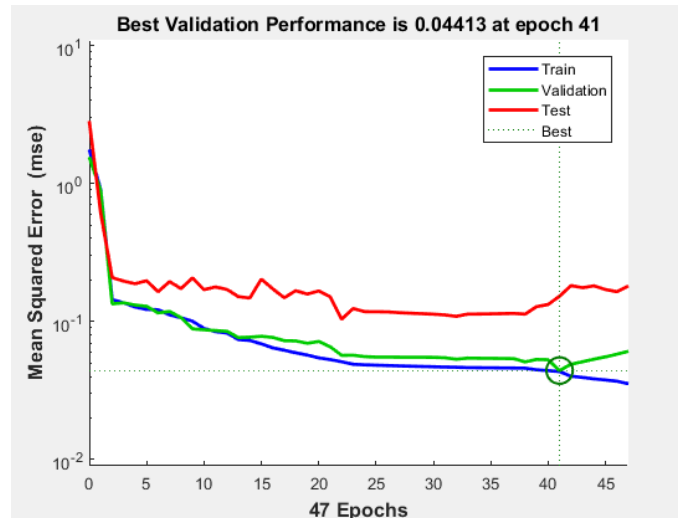


Figura 40. Gráfica de evaluación del desempeño de la Tabla 18

En la Figura 40 se visualiza el rendimiento muy bueno, ya que está cercano al 0. Aunque no es óptimo las líneas demuestran el comportamiento del entrenamiento, validación y pruebas.

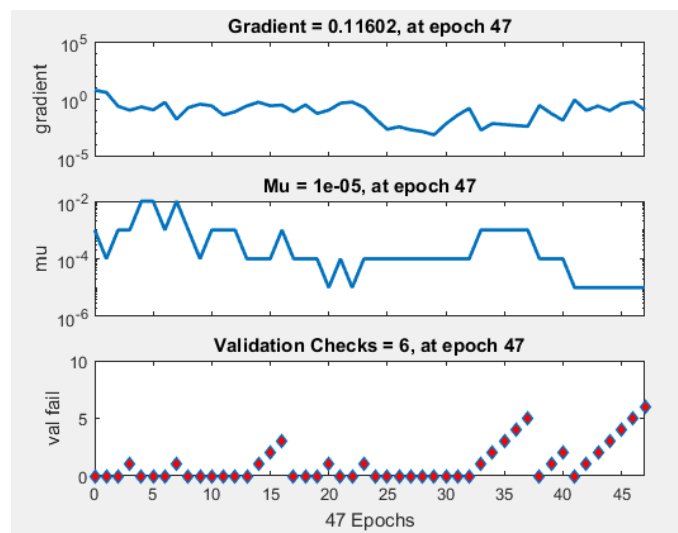


Figura 41. Gráfica del estado de entrenamiento de la Tabla 18

En la Figura 41 se visualiza el grado de ajuste entre los datos de aprendizaje y los de prueba, alcanza un valor de fallo mínimo.

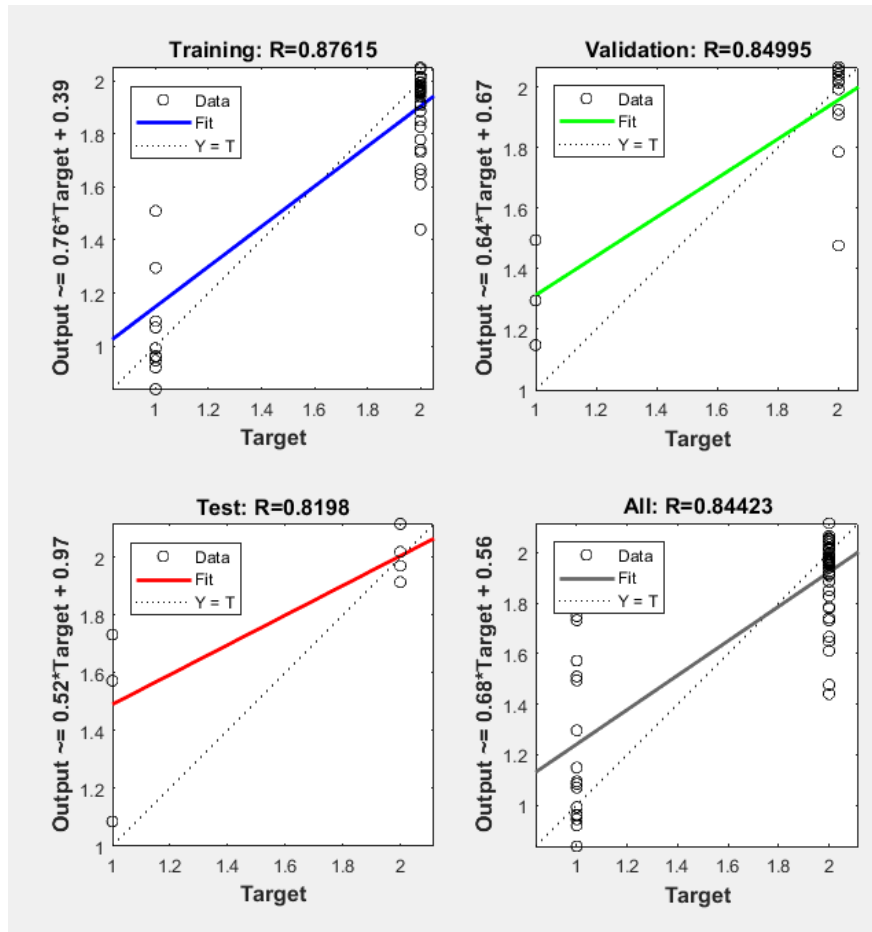


Figura 42. Gráfica de regresión del entrenamiento de la Tabla 21

La Figura 42 no muestra cercanía con el ángulo de 45° , el resultado es de 84% de eficacia del modelo. Lo que no lo hace cercano a nuestro objetivo, cuya efectividad sea mayor al 90%.

4.5.2.5 Mejor resultado con 21 muestras

Con las muestras de la Tabla 23 se probó con diferente división, se tomaron aleatoriamente 14 muestras para entrenamiento (60%), 4 muestras para validación (20%) y 4 muestras para pruebas (20%), con 10 neuronas ocultas. El algoritmo de entrenamiento con aprendizaje no supervisado trainlm. El resultado de la gráfica de rendimiento es:

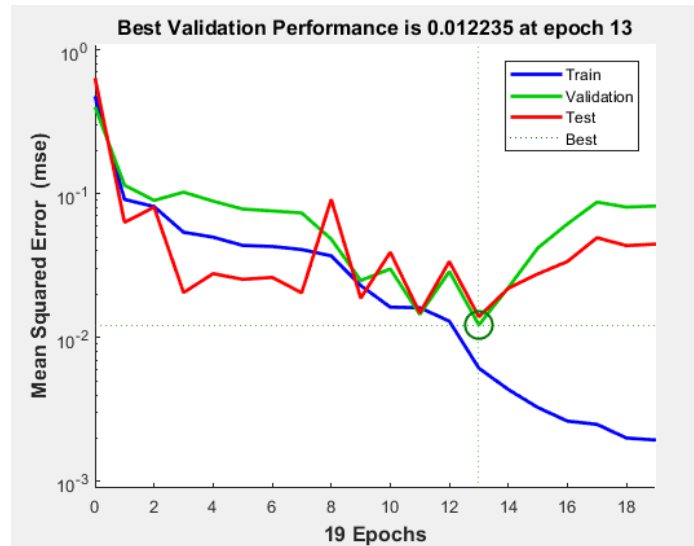


Figura 43. Gráfica de evaluación del desempeño de la Tabla 23

El desempeño en la prueba de la Figura 43 es bastante bueno, el resultado del error es cercano a 0, aunque se observa un ligero bajo ajuste en el entrenamiento.

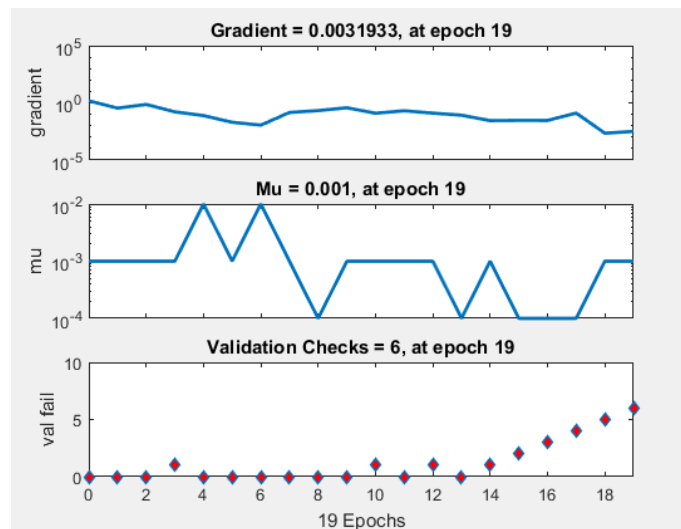


Figura 44. Gráfica del estado de entrenamiento de la Tabla 23

En la Figura 44 se visualiza el grado de ajuste entre los datos de aprendizaje y los de prueba, alcanza un valor de fallo mínimo. Aunque el error aumenta en el período 14.

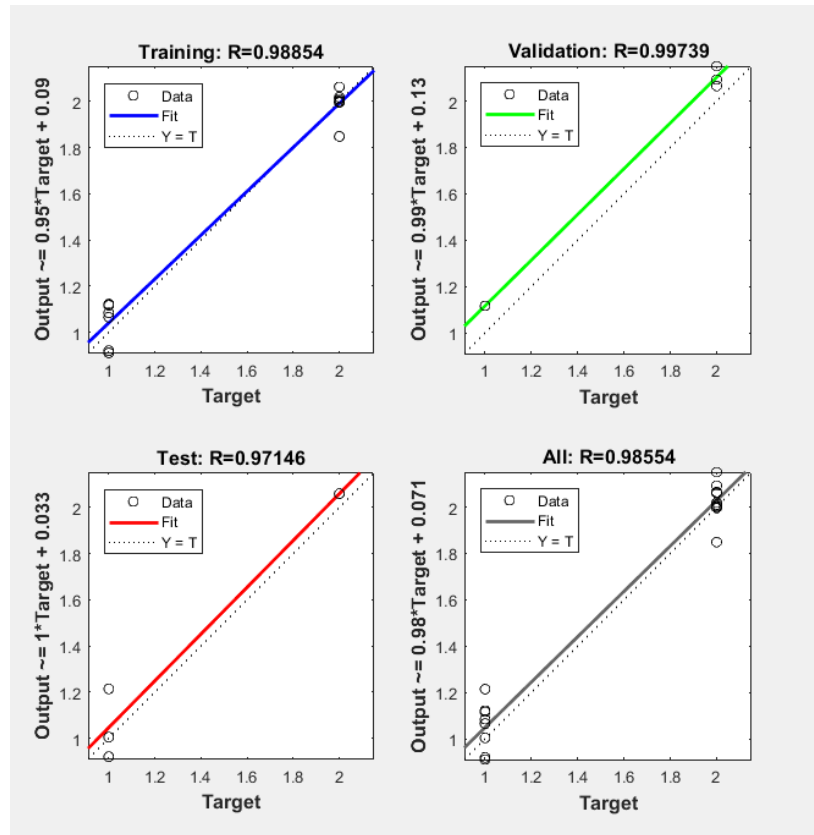


Figura 45. Gráfica de regresión del entrenamiento de la Tabla 23

La Figura 45 muestra cercanía con el ángulo de 45° en los 4 resultados, y cercanos a la unidad. El resultado es superior al 98% de efectividad, por lo que el modelo muestra que con los datos balanceados se logra el objetivo.

Después se procede a simular con los datos de prueba, y se visualizan los errores, además se extrae el error cuadrático medio. Los datos antes mencionados se exportan al workspace de Matlab y posteriormente a Excel para su tratamiento, con esos datos se compara la salida deseada contra la salida resultante.

1	error													
2	0.01768129	-0.19633956	-0.04903193	1.14E-02	1.77E-02	-1.96E-01	-0.04903193	-2.81E-01	-0.2900299	-7.88E-02	0.49456234	1.05E-02	0.0292651	-0.02593861
3	output													
4	Deseado	1	1	1	1	1	1	1	1	1	1	2	2	2
5	Resultante	0.98231871	1.19633956	1.04903193	0.98860569	0.98231871	1.19633956	1.04903193	1.2808511	1.2900299	1.07877992	0.50543766	1.98946963	1.9707349
6		98.2318714	20.8302957	104.903193	98.8605692	98.2318714	119.633956	104.903193	128.08511	129.00299	107.877992	50.5437656	99.4734817	98.5367451
7	TEV	98.9415893												

Figura 46. Comparación salida deseada contra la resultante y el porcentaje de precisión del experimento de la Tabla 23

En la Figura 46 se muestra en la fila 4 los targets de la Tabla 23 como resultado deseado, la fila 5 el resultado que se obtuvo después del entrenamiento de la red neuronal del experimento (output), la fila 6 es la resta entre el valor deseado y el valor resultante. La fila

7, es el promedio de los valores de la fila 6 porcentuado. Se le asignó el nombre de TEV del inglés total error valuation, en español evaluación total del error.

El MSE se calcula para precisar los datos resultantes de la red neuronal de la siguiente forma:

$$\text{MSE} = \text{mse}(\text{error}) = \text{mse}(\text{output-target}) \quad (30)$$

Se puede observar en la fila 7 el porcentaje de precisión del clasificador de la red neuronal.

4.5.3 Resultados por grupos

A continuación, se muestran 10 grupos de salidas de pruebas de precisión y error, los datos para el clasificador se tomaron de la Tabla 20, se dividieron en grupos para poder visualizar a detalle los valores. Los grupos están divididos en orden ascendente, es decir las pruebas de menor precisión a mayor precisión, la Figura 47 hasta la Figura 56 la disposición en el eje de las X corresponde al número de prueba y el eje de las Y al porcentaje alcanzado.

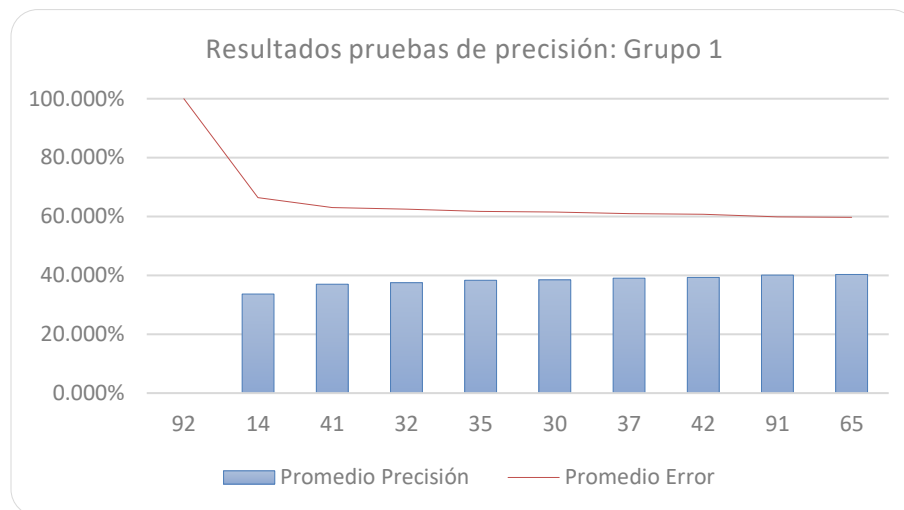


Figura 47. Resultados del experimento del grupo 1, que visualiza las pruebas 92,14,41,32,35,30,37,42,91 y 65 en orden ascendente

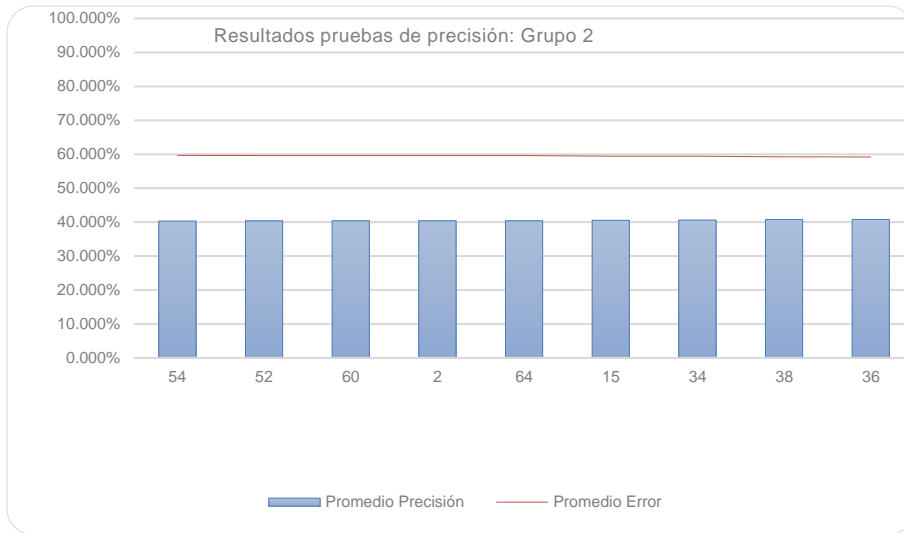


Figura 48. Resultados del experimento del grupo 2, que visualiza las pruebas 54, 52, 60, 2, 64,15,34,38 y 36 en orden ascendente

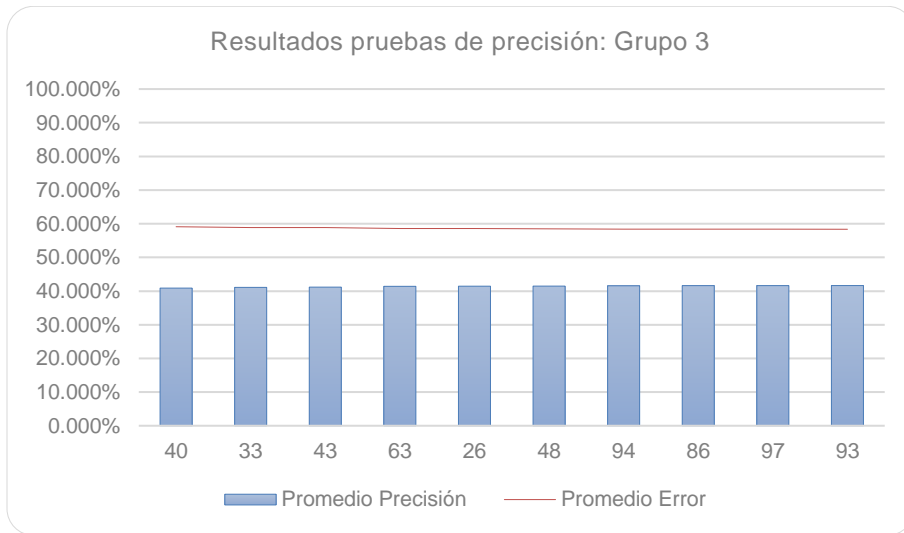


Figura 49. Resultados del experimento del grupo 3 que visualiza las pruebas 40,33,43,63,26,48,94,86,97 y 93 en orden ascendente

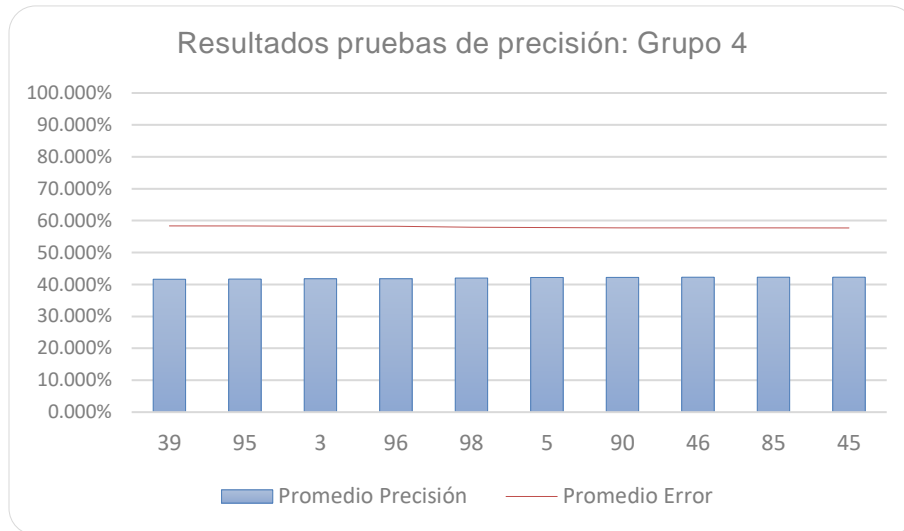


Figura 50. Resultados del experimento del grupo 4, que visualiza las pruebas 39,95,3,96,98,5,90,46,85 y 45 en orden ascendente

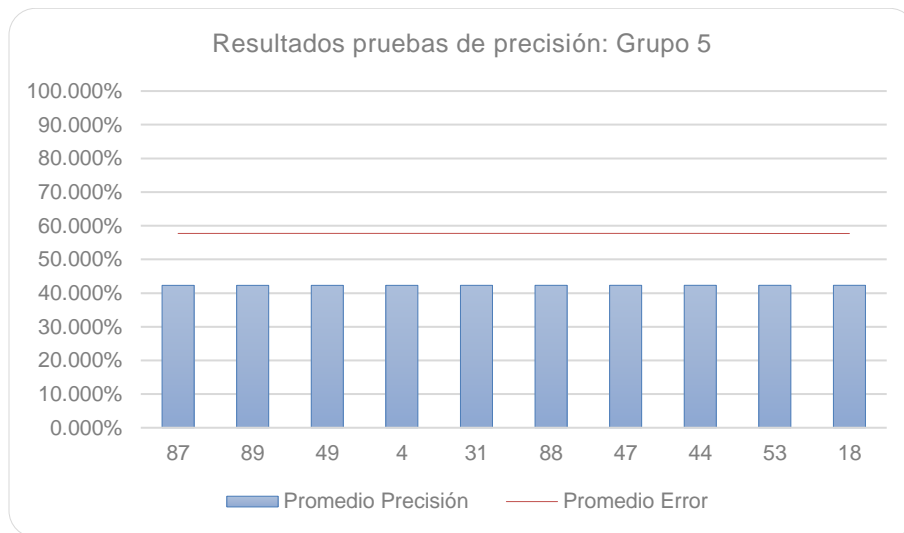


Figura 51. Resultados del experimento del grupo 5, que visualiza las pruebas 87,89,49,4,31,88,47,44,53 y 18 en orden ascendente

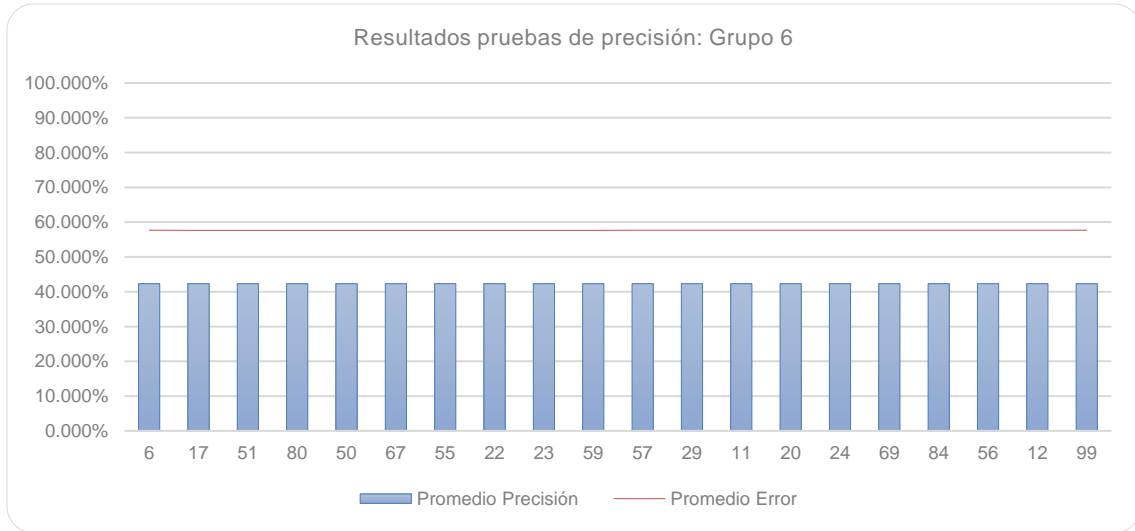


Figura 52. Resultados del experimento del grupo 6, que visualiza las pruebas 6,17,51,80,50,67,55,22,23,59,57,29,11,20,24,69,84,56,12 y 99 en orden ascendente

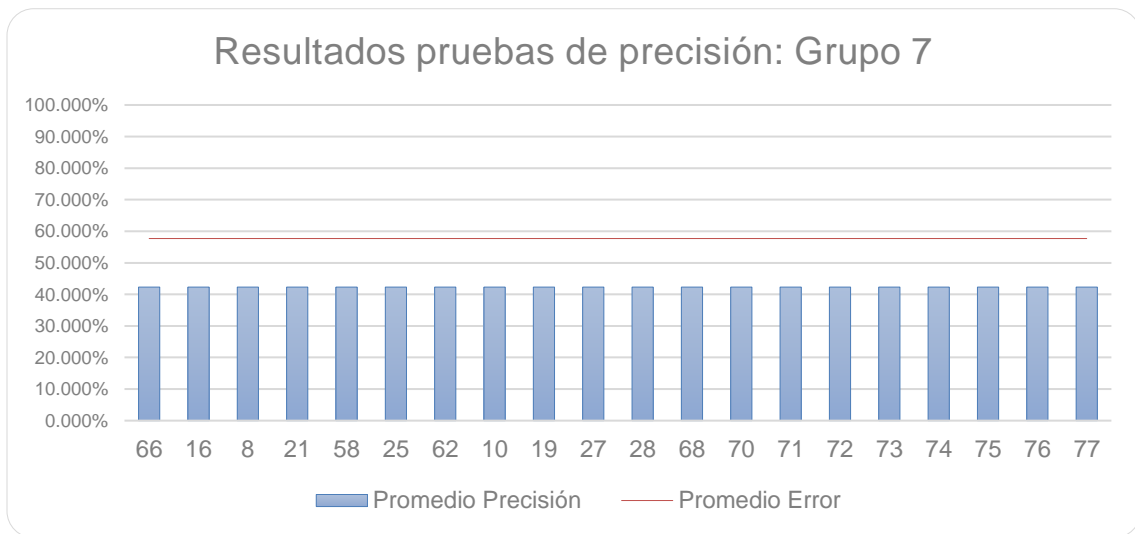


Figura 53. Resultados del experimento del grupo 7, que visualiza las pruebas 66,16,8,21,58,25,62,10,19,27,28,68,70,71,72,73,74,75,76 y 77 en orden ascendente

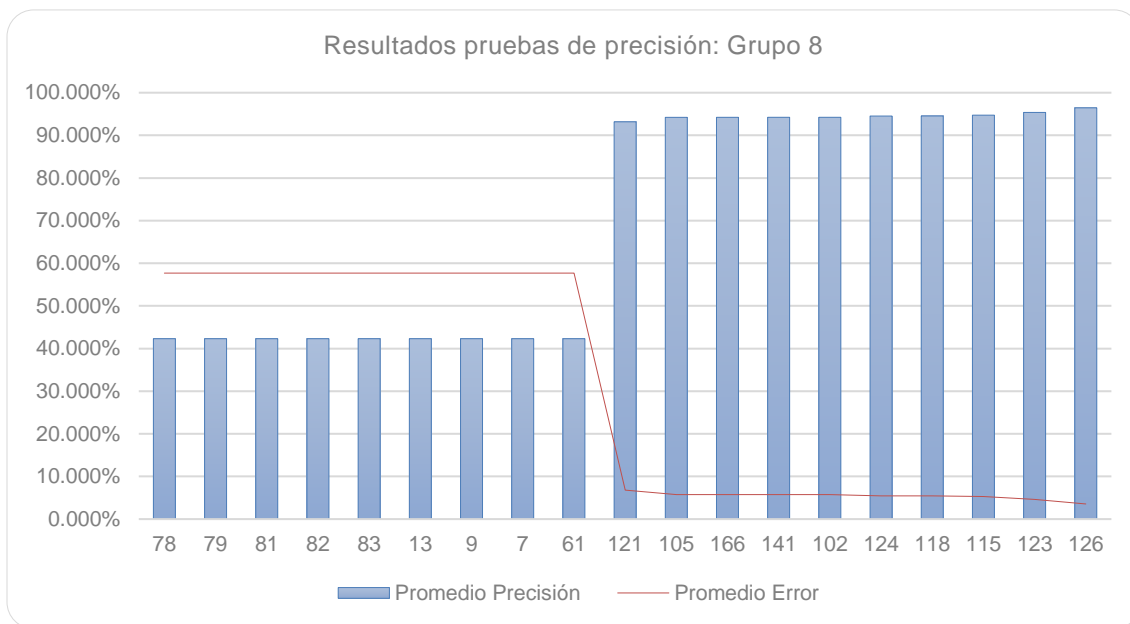


Figura 54. Resultados del experimento del grupo 8, que visualiza las pruebas 78,79,81,82,83,13,9,7,61,121,105,166,141,102,124,118,115,123 y 126 en orden ascendente

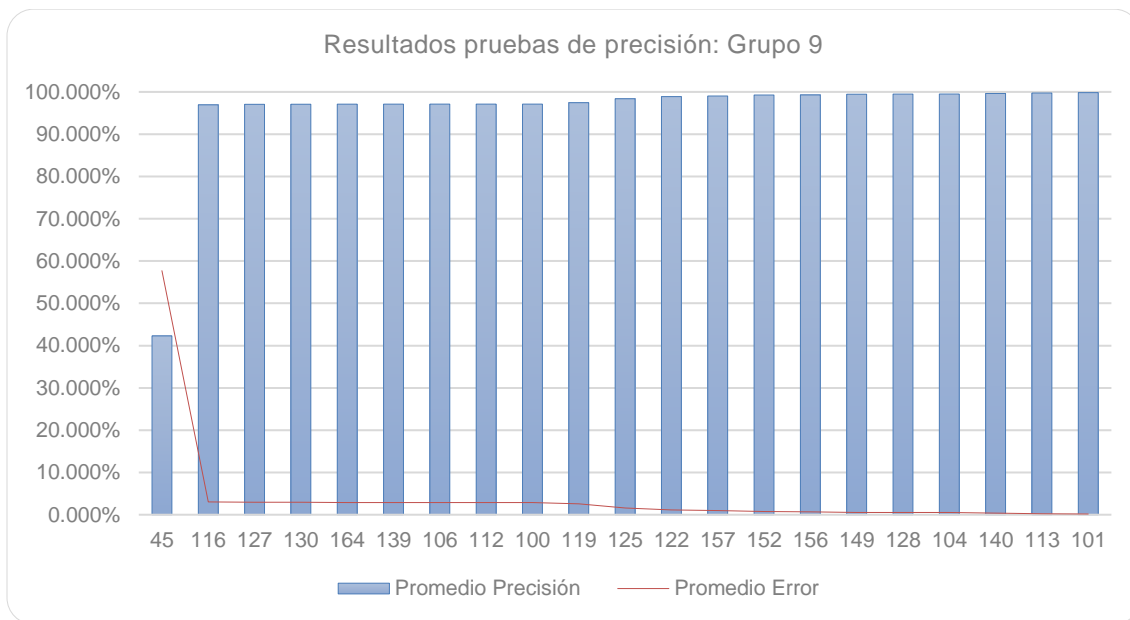


Figura 55. Resultados del experimento del grupo 9, que visualiza las pruebas 127,130,164,139,106,112, 100,119,125,122,157,152,156,149,128,104,140,113,101 y 126 en orden ascendente

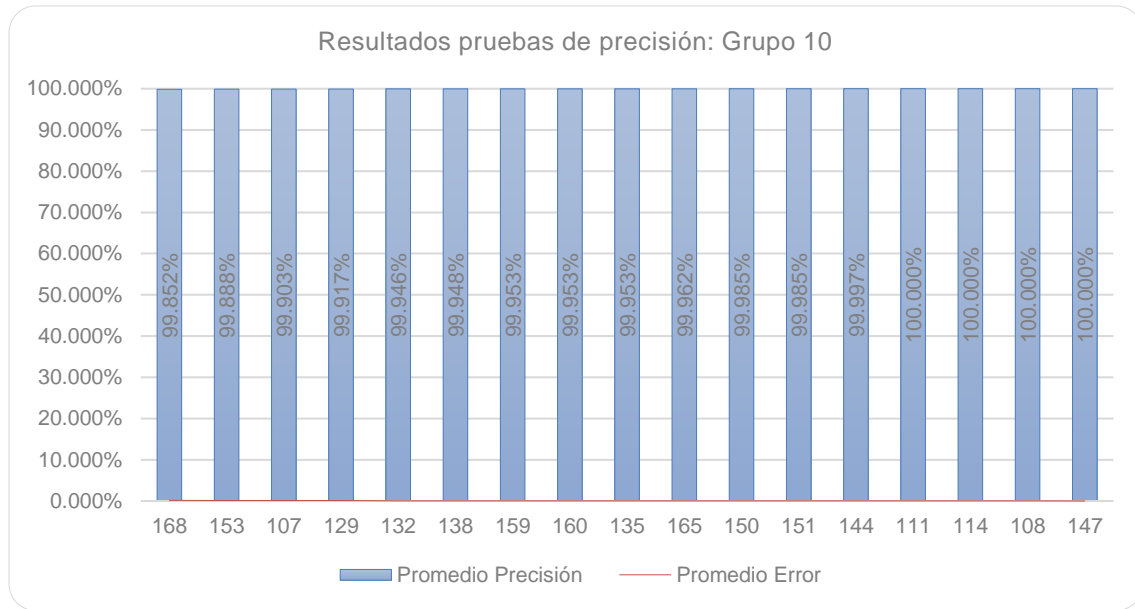


Figura 56. Resultados del experimento del grupo 10, que visualiza las pruebas 168,153,107,129,132,138,159,160,135,165,150,151,144,111,114,108 y 147

Como se observa en la Figura 56, las pruebas 111, 114, 108 y 147 salieron satisfactorias para las configuraciones dadas. Se concluye que dichas pruebas cuentan con un 98% de precisión. Sin embargo, las pruebas a pesar de ser satisfactorias por la precisión, tiene baja confiabilidad por la cantidad de muestras sanas y enfermas tomadas. La Tabla 24 además muestra las pruebas para 22 y 42 muestras sombreadas con color rosa, se resalta que las pruebas son completamente confiables, ya que como se vió en la Tabla 21 y Tabla 23, los datos están balanceados y se procede a determinar que el mejor resultado lo obtuvieron estas pruebas.

Número de Prueba	Tipo de Red Neuronal	Función de Entrenamiento	Función de desempeño	Función de transferencia	Precisión
108	Feed-forward backprop	TrainCGF	LearnGDM	TanSIG	99%
111	Feed-forward backprop	TrainCGP	LearnGD	TanSIG	99%
114	Feed-forward backprop	TrainCGP	LearnGDM	TanSIG	99%
147	Feed-forward backprop	TrainOSS	LearnGD	TanSIG	99%
22 muestras	Feed-forward backprop	TrainLM	LearnGD	TanSIG	98%
42 muestras	Feed-forward backprop	TrainLM	LearnGD	TanSIG	89%

Tabla 24. Parámetros utilizados con mejor rendimiento de las pruebas 108, 111, 114 y 147, así como para 44 y 42 muestras.

La Tabla 24 desglosa las configuraciones de las pruebas 108,111,114 y 147 notando que todas las pruebas usan la misma función de transferencia (tansig). Esta función es una buena compensación para las redes neuronales, donde la velocidad es importante y la forma exacta de la función de transferencia no lo es. En el anexo 4 se detalla la red neuronal feed-forward backprop con las funciones en combinación.

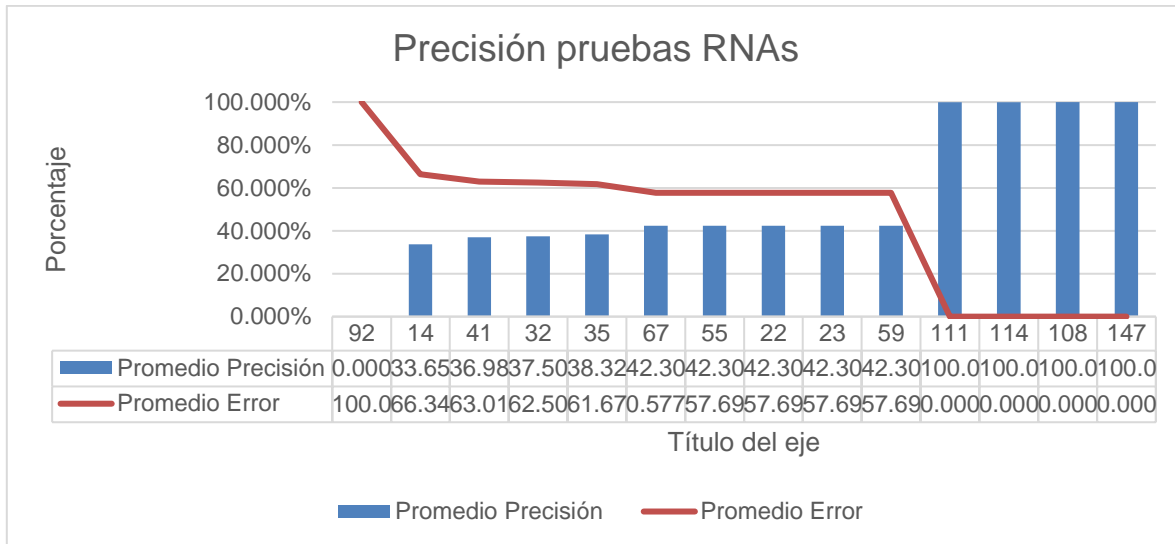


Figura 57. Figura comparación precisión de las pruebas más bajas 92,14,41,32,35, las pruebas medias 67,55,22,23,59, y las pruebas óptimas 111,114,108 y 147

La Figura 57 hace una comparación de las pruebas de menor precisión, de precisión media y de precisión alta, destacando solo algunas pruebas para su análisis. A continuación en la Tabla 25, se muestran las diferentes configuraciones de las pruebas mostradas en la Figura 51. La red cascade forward backpropagation es el tipo de red con menor incidencia, al igual que la función de transferencia purelin.

	Tipo de red neuronal	Función de entrenamiento	Función de rendimiento	Función de transferencia
Prueba 92	Feed-Forward Backprop	TrainBR	LearnGD	PureLIN
Prueba 14	Cascade Forward Backprop	TrainCGB	LearnGD	PureLIN
Prueba 41	Cascade Forward Backprop	TrainGDM	LearnGDM	PureLIN
Prueba 32	Cascade Forward Backprop	TrainGD	LearnGD	PureLIN
Prueba 35	Cascade Forward Backprop	TrainGD	LearnGDM	PureLIN
Prueba 67	Cascade Forward Backprop	TrainR	LearnGD	LogSig

Prueba 55	Cascade Forward Backprop	TrainLM	LearnGD	LogSig
Prueba 22	Cascade Forward Backprop	TrainCGF	LearnGDM	LogSig
Prueba 23	Cascade Forward Backprop	TrainCGF	LearnGDM	PureLIN
Prueba 59	Cascade Forward Backprop	TrainLM	LearnGDM	PureLIN

Tabla 25. Parámetros utilizados con mejor rendimiento de las pruebas bajas 92,14,41,32,35, las pruebas con resultado medio 67,55,22,23,59

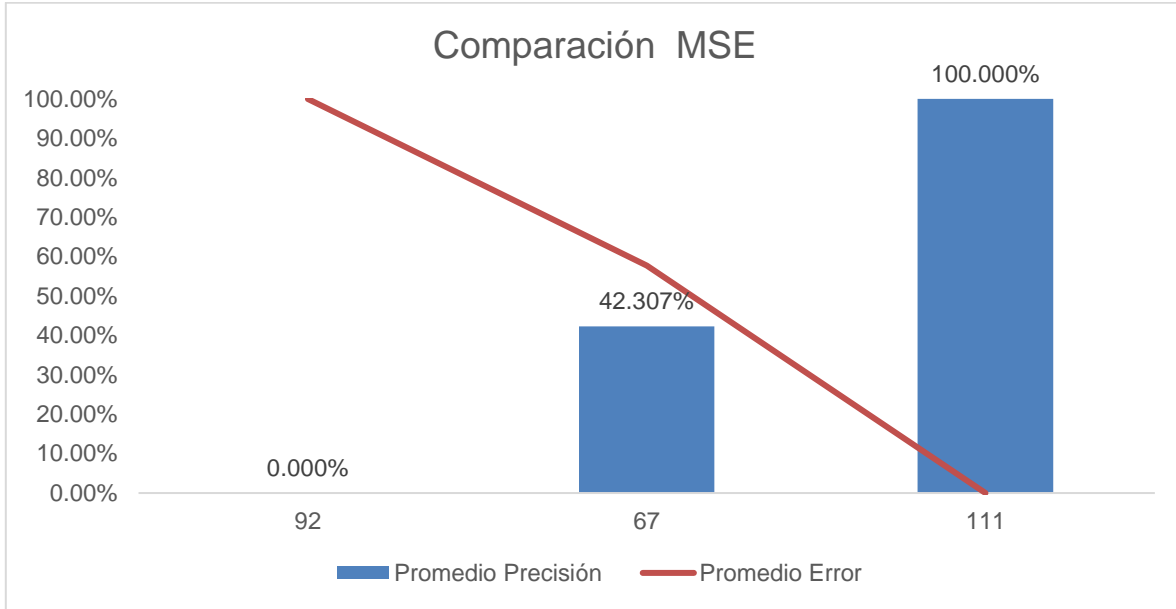


Figura 58. Figura comparación MSE de las pruebas 92, 67 y 111.

La Figura 58 muestra a detalle el resultado de los entrenamientos: la menor precisión la obtuvo la prueba 92, la prueba 67 obtuvo una precisión media y el resultado de las pruebas 108,111,147 y 114 como la precisión más alta, seleccionando solo la prueba 111 de la clasificación de la Tabla 20.

Conclusiones

El algoritmo feed-forward backpropagation es altamente efectivo y utilizado en diversas aplicaciones y propósitos, así mismo en la mayoría de las referencias que se dieron en este documento de tesis utilizan el algoritmo. A través del tiempo y de los avances médicos nos dimos cuenta de que existen diversos estudios de clasificación de señales, usando métodos como el análisis en tiempo y frecuencia. En este trabajo se logró obtener 19 características importantes de las señales EMG para su clasificación en diferentes desordenes musculares. El resultado de los experimentos nos guio a organizar mejor los datos para la entrada al clasificador.

Esta tesis generó una contribución para la investigación y desarrollo de los sistemas para el diagnóstico de enfermedades neuromusculares, ya que diferentes autores utilizan pocas características extraídas de la señal. Es importante mencionar que los datos que se obtuvieron son reales, y la aplicación para este tema fue el diagnóstico.

Se cuenta con más del 90% de precisión de la clasificación en las pruebas 107,108, 111, 114 y para 22 muestras. Lo que resultan atractivas dichas configuraciones para el futuro procesamiento de las señales EMG. Las pruebas con 22 y 42 muestras son completamente confiables, los datos están balanceados y se procede a determinar que el mejor resultado lo obtuvieron estas pruebas, con un 98% de efectividad para la prueba con 22 muestras y 89% de efectividad con 42 muestras.

Se tiene la hipótesis de que los resultados menos acertados se dieron dado que cuando se configuró la red se puede o no ingresar las neuronas con las que puede trabajar, es decir, se le indica cuales son parámetros objetivos y límite superior e inferior. Debido al conjunto de datos que se tiene, menos de 10 neuronas no ajustaba el modelo, más de 10 neuronas se ajustaba mejor, en específico con 10 neuronas está comprobado el buen funcionamiento de la red con los datos.

En el futuro, se puede usar la técnica de la extracción de características en combinación con otros métodos de clasificación como Support Vector Machine para crear una mayor robustez.

Como se observa se logra una precisión en mayor en 6 redes neuronales, usando el algoritmo Feed Forward Back Propagation con las configuraciones de la Tabla 24. La función de activación se usa para convertir la entrada en una salida delimitada. Para redes neuronales multicapa las funciones de transferencia son diferenciables para realizar aproximación de funciones, asociación de patrones y clasificación de patrones.

Trabajo Futuro

Se recomienda:

- Queda abierta la posibilidad de mejorar el clasificador, haciendo pruebas más exactas desglosando neuropatía, miopatía y sano (3 resultados deseados) comenzando con los datos que ya se almacenan, crear un clasificador con pocas señales y luego probarlos con nuevas y observar su comportamiento.
- Crear un clasificador para cada uno de los diferentes músculos afectados, incluyendo los frontales y dorsales.
- Validar los instrumentos de adquisición comparando el resultado y verificar que instrumento mejora la señal.

Por otra parte, es de interés elaborar juntamente con un neurólogo un sistema embebido para la detección oportuna de enfermedades neuromusculares y se podría a su vez alimentar la red neuronal con datos de enfermedades que comúnmente ataca a los mexicanos por su actividad laboral, sus enfermedades hereditarias y sus condiciones de riesgo.

Crear un módulo físico de adquisición de señales electromiográficas eficaz y de bajo costo.

Se considera en un futuro inmediato la escritura de un artículo científico que muestre el resultado de los experimentos obtenidos en una revista internacional.

Referencias

- A, R., M, N., R, M., D, F., I, C., & S., G. (2000). Geometrical factors in surface EMG of the vastus medialis and lateralis muscles. *J Electromyogr Kinesiol* (pp. 327-336). ELSEVIER.
- A.M., B., M, T., & V, B. (2003). Normalisation of gait EMGs: a re-examination. *Journal Electromyography Kinesiology*, 519-532.
- Ahmad, I., & Ansari, F. (2012). A review of EMG recording technique. *International journal of engineering science and technology*.
- Ahsan, M., Ibrahimy, M., & Khalifa, O. (2011). Hand motion detection from EMG signals by using ANN based classifier for human computer interaction. *Fourth international conference on modeling, simulation and applied optimization* (pp. 1-6). Kuala Lumpur: IEEE.
- Ahsan, R., Ibrahimy, M., & Khalifa, O. (2012). The use of artificial neural network in the classification of EMG signals. *3rd FTRA International conference on mobile, ubiquitous, and intelligent computing* (pp. 225-229). Vancouver: IEEE.
- Albers, J. W. (2000). Chapter 7: Occupational, biologic, and environmental toxic neuropathies. In P. D. Donofrio, *Textbook of Peripheral Neuropathy* (pp. 87-105). New York: Demos medical.
- Alva, C. C. (2012). *Procesamiento de señales de electromiografía superficial para la detección de movimiento de dos dedos de la mano*. Lima, Perú: Universidad Ricardo Palma. Retrieved from http://repositorio.urp.edu.pe/bitstream/handle/urp/416/Alva_ca.pdf?sequence=1&isAllowed=y
- Álvarez Fiallo, R., Santos Anzorandia, C., & Medina Herrera, E. (2007, Enero). *Scielo*. Retrieved from http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0138-65572007000100008
- Angkoon Phinyomark, C. L. (2011). Application of Wavelet Analysis in EMG Feature Extraction for Pattern Classification. *Measurement Science Review*, 45-52.
- Ankoom, P., Quaine, F., & Laurillan, Y. (2012). The relationship between anthropometric variables and features of electromyography signal for human-computer interface. In N. Ganesh R., *Applications, challenges, and advancements in electromyography signal processing* (pp. 321-354). United states of america: Medical Information Science Reference.
- Benoit, D., M., L., G., C., & A., L. (2003). The clinical significance of electromyography normalization techniques in subjects with anterior cruciate ligament injury during treadmill walking. *Elsevier*, 56-63.

- Bodruzzaman, M., Zein-Sabatto, S., Marpaka, D., & Kari, S. (1992). Neural network-based classification of electromyographic (EMG) signal during dynamic muscle contraction. *Proceedings IEEE Southeastcon '92, Birmingham* (pp. 99-102). AL,USA: IEEE.
- Boostani, R., & Moradi, M. (2003). Evaluation of the forearm EMG signal features for the control of a prosthetic hand. *Physiological measurement*, 309-319.
- Burden, A., & R., B. (1999). Normalisation of EMG amplitude: an evaluation and comparison of old and new methods. *Med Eng Phys*, 247-257.
- Chen, K.-H. S., & Chen, R. (2020). Principles of Electrophysiological Assessments for Movement Disorders. *Journal of movement disorders*, 27-38.
- Chugani, H. (1998). Biological basis of emotions: Brain systems and brain development. *Pediatrics*, 1225-1299.
- Cifuentes, G. I. (2012). Extracción de características y clasificación de señales electromiográficas utilizando la transformada hilbert-huang y redes neuronales. *INAOE*.
- Clinic, M. (2019). *Mayo clinic*. Retrieved from [<https://www.mayoclinic.org/es-es/tests-procedures/emg/about/pac-20393913>]
- Cocero, O. E., & Recuero, F. E. (1971). *Introducción a la electromiografía y a la conducción eléctrica del nervio periférico*. Madrid: Graphia.
- Deepika, C., & Dr. A.k, W. (2011). Application of wavelet analysis in EMG feature extraction for pattern classification. *Measurement science review*, 45-52.
- Deepika, C., & Wadhvani, D. A. (2012). Analysis of different EMG signals by segmentation, classification and feature extraction phase. *International journal of engineering science & technology*, 4(4), 1680.
- Demuth, H., & Mark, B. (2004). *Neural network toolbox*. Retrieved from www.mathworks.com
- Dua, D., & Graff, C. (2019). *UCI Machine learning repository*. Retrieved from University of california, school of information and computer science: <http://archive.ics.uci.edu/ml>
- Durán Acevedo, C. M. (2014). Desarrollo de un sistema integrado para la clasificación de señales electromiográficas. *Excellence in engineering to enhance a country's productivity*. Guayaquil, Ecuador.
- Elamvazuthi, I., Duy, N., Zulfiqar, A., S.W. , S., Khan, M., & Parasuraman, S. (2015). Electromyography (EMG) based classification of neuromuscular disorders using multi-layer perceptron. *IEEE International symposium on robotics and intelligent sensors*, 223-228.

- Exchange, S. (2015, 10 12). *Stack exchange*. Retrieved from <https://stats.stackexchange.com/questions/19048/what-is-the-difference-between-test-set-and-validation-set>
- Foundation, A. (2020). *khan Academy*. Retrieved from <https://es.khanacademy.org/science/biology/human-biology/neuron-nervous-system/a/the-synapse>
- Galván, M. I., & Valls, J. M. (2018). *Universidad Carlos III de Madrid*. Retrieved from <http://ocw.uc3m.es/ingenieria-informatica/redes-de-neuronas/transparencias/Explicacion%20Tema5.pdf>
- García, S. A. (2018). *Inteligencia artificial: Fundamentos, práctica y aplicaciones*. Alfaomega.
- Goen, A. (2014). Classification of EMG Signals for Assessment of Neuromuscular Disorders. *International Journal of Electronics and Electrical Engineering Vol. 2, No. 3, 242-248*.
- González, G. R. (2009). *Análisis de series de tiempo con redes neuronales*. San Luis Potosí, México: UASLP.
- González, I. J. (2012). *Reconocimiento de señales mioeléctricas mediante redes neuronales artificiales [Tesis de maestría]*. San Luis Potosí: UASLP.
- González, I. J., Soubervielle, M. C., Vital, O. O., & Pérez, G. H. (2012). EMG pattern recognition system based on neural networks. *11th Mexican International Conference on Artificial Intelligence* (pp. 71-74). San Luis Potosí, México: IEEE.
- Guerrero, M. J. (2010-2011). *Universidad de Valencia*. Retrieved from Open Course Web: http://ocw.uv.es/ingenieria-y-arquitectura/1-5/ib_material/IB_T2_OCW.pdf
- Hamui, S. L., Fuentes, G. R., Aguirre, H. R., & Ramírez, d. l. (2015). Un estudio de satisfacción con la atención médica. *Expectativas y experiencias de los usuarios del sistema de salud en México*, 109.
- IENSA. (2020). *Instituto de especialidades neurológicas*. Retrieved from <http://www.iensa.es/especialidades/neurofisiologia/item/25-electromiografia-emg/25-electromiografia-emg>
- INEGI. (2015). *INEGI Cuenta*. Retrieved from <http://cuentame.inegi.org.mx/poblacion/habitantes.aspx?tema=P>
- Institut, S. E. (2011, Julio). *Sport Elec Institut*. Retrieved from <https://www.sport-elec.es/zoom/colocacion-de-los-electrodos-segun-los-musculos>
- Intermedica. (2009). *Intermedica*. Retrieved from <http://www.intermedica.com.ec/es/producto/technomed.php>

- Irfan, M., Sudharsan, N., Santhanakrishnan, S., & B. , G. (2011). A comparative study of EMG and MMG signals for practical applications. *International conference on signal, image processing and applications* (pp. 106-110). Singapore: IACSIT Press.
- ISEK. (2015). *Journal of Electromyography and Kinesiology*. Retrieved from <https://isek.org/wp-content/uploads/2015/05/Standards-for-Reporting-EMG-Data.pdf>
- Jarušek , R., & Kocian, V. (2018). Artificial intelligence algorithms for classification and pattern recognition. In I. Global (Ed.), *Computer vision: Concepts, methodologies, tools, and applications* (Vol. 4, pp. 350-376). Czech Republic: Information resources management association.
- Jingwei, T., Abdul, R. A., Norhashimah, M. S., & Weihown, T. (2019). EMG feature selection and classification using a pbest-guide binary particle swarm optimization. *Computation*, 1-20.
- Karlik, B. (2014). Machine learning algorithms for characterization of EMG signals. *International Journal of Information and Electronics Engineering* (pp. 189-194). Turkey: IJIEE.
- Karyakarte, S., & Savant, P. I. (2019). Pattern recognition process, methods and applications in artificial intelligence. *International research journal of engineering and technology* (pp. 1162-1166). India: IRJET.
- Katsutoshi, K., Seiji, S., Koji, O., & Takao, T. (1993). A discrimination system using neural network for EMG-controlled prostheses integral type of EMG signal processing. *Proceedings of the international conference on intelligent robots and systems*, 1750-1755.
- Khan, M. U., Aziz, S., Bilal, M., & Aamir, M. B. (2019). Classification of EMG signals for assessment of neuromuscular disorders using empirical mode decomposition and logistic regression. *International Conference on Applied and Engineering Mathematics (ICAEM)* (pp. 237-243). Taxila, Pakistan: IEEE.
- Khan, M., Aziz, S., Bilal, M., & Aamir, M. (2019). Classification of EMG signals for assessment of neuromuscular disorder using empirical mode decomposition and logistic regression. *International Conference on Applied and Engineering Mathematics (ICAEM)* (pp. 237-243). Taxila, Pakistan: IEEE.
- LaMontagne, Anthony, Keegel, & Tessa. (2012). Reducing stress in the workplace: An evidence review. *VicHealth*.
- Laxmi, S., & Sangeeta, B. (2012). Online EMG signal analysis for diagnosis of neuromuscular disease by using PCA and PNN. *International journal of engineering science & technology* (pp. 4453-4458). Lagos Nigeria: IJEST.

- M., A., G., G., & M., F. (2012). Classification of EMG signals through wavelet analysis and neural networks for controlling an active hand prosthesis. *IEEE Proceedings of the International Conference on Rehabilitation Robotics*, 531-536.
- M. Bodruzzaman, S. Z.-S. (1992). Neural network-based classification of electromyographic emg signal during dynamic muscle contraction. *IEEE*, 99-102.
- M. Zardoshti, K., & Afshordi, A. (1995). Classification of chromosomes using higher-order neural networks. *IEEE International conference on neural networks Vol. 5*. IEEE Xplore.
- MATLAB. (2004). *Neural Network Toolbox*. Natick, MA: The MathWorks, Inc.
- Meneses, A. (2005). *Dalcame*. Retrieved from <http://www.dalcame.com/emg.html>
- Mulroy, S. &. (2003). Use of cluster analysis for gait pattern classification of patients in the early and late recovery phases following stroke. *Gait and posture*, 114-125.
- National institute of neurological disorders, s. (2015). *Emglab*. Retrieved from <http://www.emglab.net>
- neurología, S. e. (2004, Julio 7). *Algoritmos diagnósticos en las miopatías*. Retrieved from Grupo de estudio de enfermedades neuromusculares: http://www.sen.es/pdf/neuromuscular/miopatias_hereditarias.pdf
- Onishi, H., R., Y., Oyama, M., Akasaka, K., Ihashi, K., & Handa, Y. (2002). EMG-angle relationship of the hamstring muscles during maximum knee flexion. *Elsevier: Journal of Electromyography and Kinesiology*, 399-406.
- Osuna , S., González, J., & Ilzarbe, A. (2012). *Procesado de señales EMG en trastornos neuromusculares*. Barcelona: Universidad Politecnica de Cataluña.
- Patcharin, A., Sivarit, S., & Kosin, C. (2016). An EMG-based feature extraction method using a normalized weight vertical visibility algorithm for myopathy and neuropathy detection. *Springer Plus*, 5:2101.
- Patidar, M., Jain, N., & Parikh, A. (2013). Classification of normal and myopathy EMG signals using BP neural network. *International journal of computer applications*, 12-16.
- Pauk, J. (2008). Different techniques for EMG signal processing. *Journal of Vibroengineering*, 571-576.
- Phinyomark, A., Thongpanja, S., Hu, H., Phukpattaranont, P., & Limsakul, C. (2013). The usefulness of mean and median frequencies in electromyography analysis. In IntechOpen, *Computational intelligence in electromyography analysis - A perspective on current applications and future challenges* (pp. 195-220). Book Citation Index.
- Physiology, M. L. (2018). *National Institute of General Medical Sciences (NIGMS)*. Retrieved from Physionet: <https://physionet.org/lightwave/?db=emgdb/1.0.0>

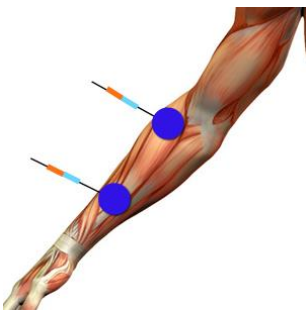
- Powar, O. S., & Chemmangat, K. (2019). Comparison of pre-processing filters on the performance of sEMG based pattern recognition. *International conference on power electronics applications and technology in present energy scenario (PETPES)* (pp. 1-6). Mangalore, India: IEEE.
- RAE. (2019, 01). *Real Academia Española*. Retrieved from <https://dle.rae.es>
- Raez, M., Mohd, Y., & Faisal. (2006). Techniques of EMG signal analysis: detection, processing, classification and applications. *Biological Procedures Online*, 11-35.
- Rainoldi, A., Melchiorri, G., & Caruso, I. (2004). A method for positioning electrodes during surface EMG recordings in lower limb muscles. *Journal of neuroscience methods* (pp. 37-43). Italia: Elsevier.
- Roetenberg, D., Buurke, J., Veltink, P., Forner Cordero, A., & Hermens, H. (2003). Surface electromyography analysis for variable gait. *ELSEVIER*, 109-117.
- Sadi-Ahmed, N., Messekher, A., Namane, S., Kedir-Talha, M., & Mekaoui, S. (2017). Automatic diagnosis of neuromuscular diseases from electromyographic (EMG) records. *5th International Conference on Electrical Engineering - Boumerdes (ICEE-B)*, 1-6.
- Sadi-Ahmed, N., Messekher, A., Namane, S., Kedir-Talha, M., & Mekaoui, S. (2017). Automatic diagnosis of neuromuscular diseases from electromyographic (EMG) records. *5th International Conference on Electrical Engineering - Boumerdes (ICEE-B)* (pp. 1-6). Boumerdes: IEEE.
- sen. (2004, Julio 7). *Algoritmos diagnósticos en las miopatías*. Retrieved from Grupo de estudio de enfermedades neuromusculares: http://www.sen.es/pdf/neuromuscular/miopatias_hereditarias.pdf
- SEN. (2004, 07 07). *Algoritmos diagnósticos en las miopatías*. Retrieved from Grupo de estudio de enfermedades neuromusculares: http://www.sen.es/pdf/neuromuscular/miopatias_hereditarias.pdf
- SENIAM. (2000). *Recommendations for surface EMG*.
- Singh, A., Dutta, M., & Travieso, C. (2017). Analysis of EMG signals for automated diagnosis of myopathy. *IEEE Uttar pradesh section international conference on electrical, computer and electronics (UPCON)*, 628-631.
- Solnik, DeVita, S. &, Rider, P. &, Long, P. &, Hortobágyi, B. &, & Tibor. (2008). Teager–Kaiser Operator improves the accuracy of EMG onset detection independent of signal-to-noise ratio. *Bioengineering and biomechanics* (pp. 65-68). National Library of Medicine.
- Subasi, Abdulhamit. (2019). Chapter 3 - Biomedical signal processing techniques. In *Practical guide for biomedical signal analysis using machine learning techniques: A MATLAB based approach* (pp. 89-192). Jeddah, Saudi Arabia: Academic Press.

- Tohka, J., & Ruotsalainen, U. (2012). *Pattern recognition*. (D. o. Science, Editor) Retrieved June 12, 2020, from University of Vermont: http://www.cs.tut.fi/sgn/m2obsi/courses/IPR/Lectures/IPR_Lecture_1.pdf
- Toledo, P. D., Rodríguez, R. J., Gómez, L. R., & Jauregui, C. J. (2019). Support vector machine-based EMG signal classification techniques: A review. *Applied sciences*, 1-28.
- Tsao, H., & Hodges, P. (2007). Immediate changes in feedforward postural adjustments following voluntary motor training. *Springer*, 537-546.
- Tsuji, T., Fukuda, O., Kaneko, M., & Ito, K. (2000). Pattern classification of time-series EMG signals using neural networks. *International journal of adaptive control and signal processing* (pp. 829-848). John Wiley & Sons, Ltd.
- UNAM. (2013). *Facultad de Medicina UNAM*. Retrieved from <http://www.facmed.unam.mx/bmnd/SatisfaccionAtencionMedica.pdf>
- Vacic, V. (2005). *Summary of the training functions in Matlab neural network toolbox*. Retrieved from http://alumni.cs.ucr.edu/~vladimir/cs171/nn_summary.pdf
- Weiss, J. M., Weiss, L. D., & Silver, J. K. (2004). *Easy EMG*. NY: Elsevier.
- William, R. (2019, 10 24). *University of Delaware*. Retrieved from Mathematics and Signal Processing for Biomechanics: <http://www1.udel.edu/biology/rosewc/kaap686/notes/EMG%20analysis.pdf>

Anexos

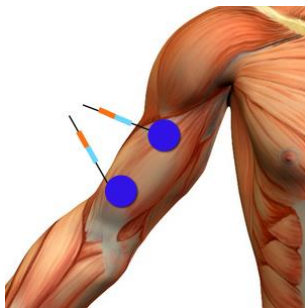
Anexo 1. Ubicación de los electrodos

Músculo supinador largo del antebrazo
(Institut, 2011)

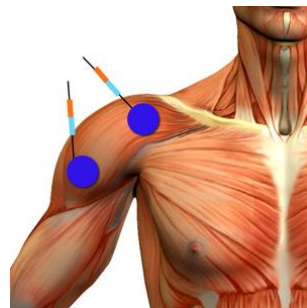


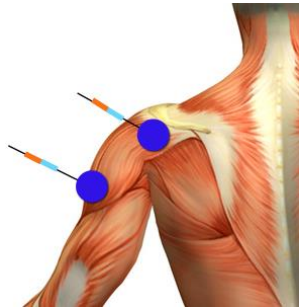
Este posicionamiento es útil para muscular y ganar volumen a través de los programas de refuerzo muscular de los aparatos. Pero también es muy útil para disminuir los efectos de los ácidos lácticos (sustancias fabricadas por los músculos después de un esfuerzo intenso o de ejercicio deportivo que provoca dolores durante las horas que siguen al esfuerzo).

Bíceps



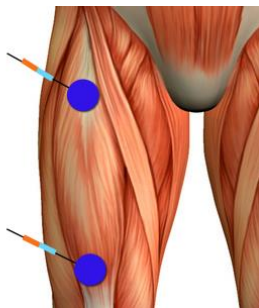
Deltoides



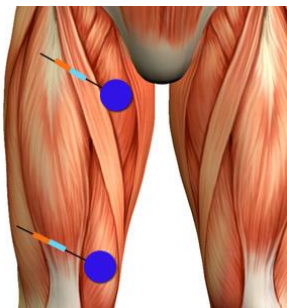


Los muslos

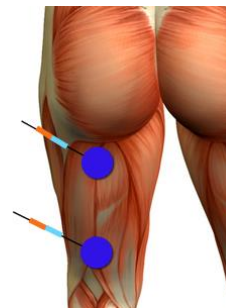
El Recto anterior



El vasto interno

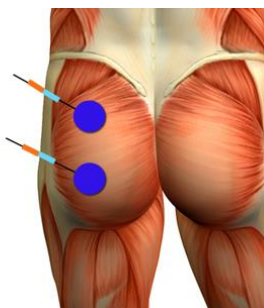


Los Bíceps (crural) del muslo



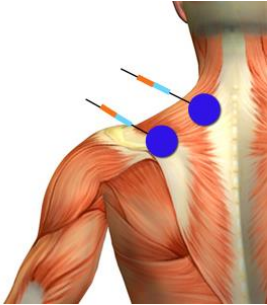
Los glúteos

El gran glúteo

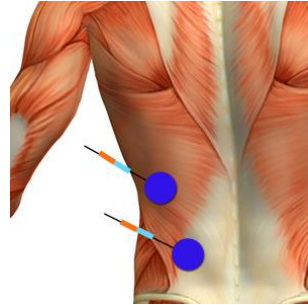


Los trapecios y dorsales

Los trapecios

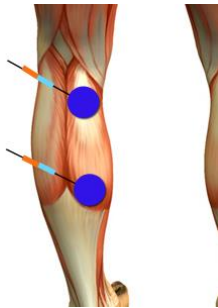


Los dorsales

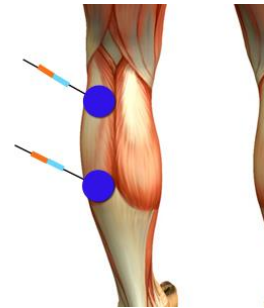


Las piernas y los gemelos

El gemelo interno

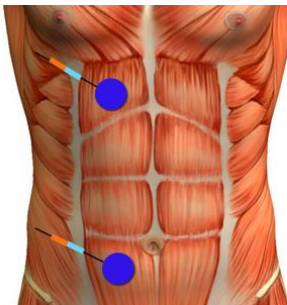


El gemelo externo

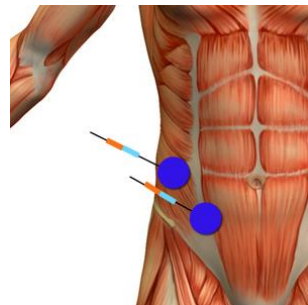


Los abdominales

El gran recto

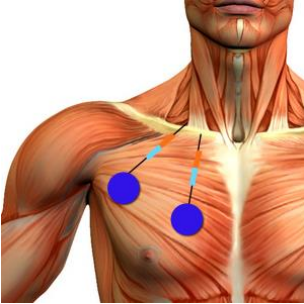


Los oblicuos

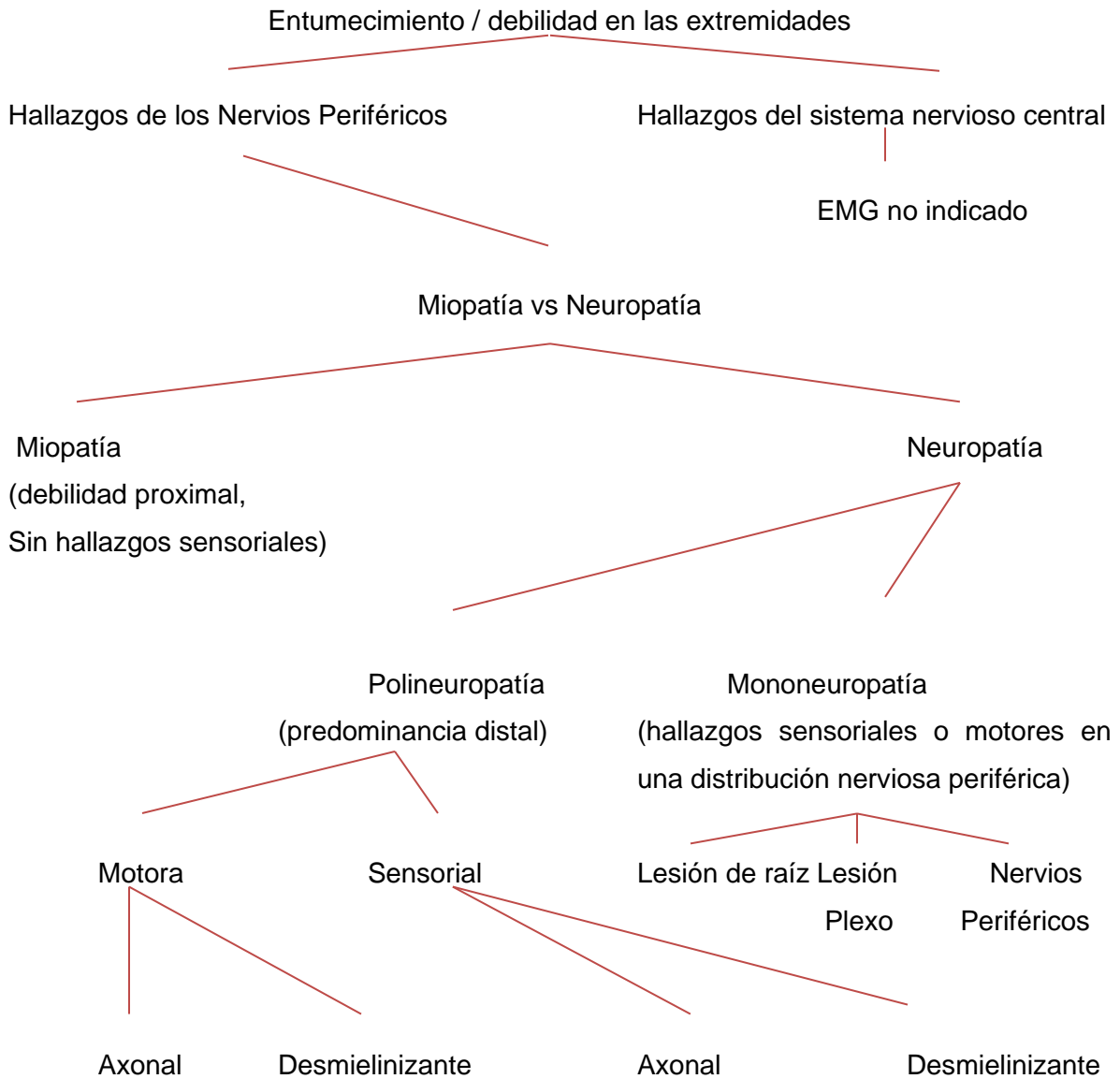


Pectorales

El gran pectoral



Anexo 2. Planificación del electro diagnóstico



Algoritmo para planificar el examen electro diagnóstico

(Weiss, Weiss, & Silver, 2004)

Anexo 3. Código fuente Matlab

Filtro notch

completenotch.m

```
function [signal_filtered, signal_fft] = complete_notch(X)

%% PROGRAM :
% Write a program in Matlab to "Load & plot EmG signal in time domain" and implement Notch
Filter to remove 50 Hz with Q fator 1
w=50/(1000/2); %w frecuencia de corte del filtro notch 50 hz
bw=w; %bandwidth
[num,den]=iirnotch(w,bw); % notch filter implementation
emg_notch=filter(num,den,X);

fft_notch=fft(emg_notch);
N1 = length(fft_notch);
fft_notch_abs= abs(fft_notch/N1);
fft_notch_abs = fft_notch_abs/max(fft_notch_abs);

signal_filtered = emg_notch;
signal_fft = fft_notch_abs;

end
```

Procesamiento de la señal

Process_Signal.m

```
function [feat, signal, signal_notch, signal_passband, signal_fft] = Process_Signal(filename)
[t1,X] = GetSignalFromFile(filename);

signal = X;

[ sig_notch, sig_notch_fft ] = complete_notch(X);
signal_notch = sig_notch;
```

% Sampling Frequency (Hz)

Fs = 1000;% Frecuencia de muestreo tiene que ser por lo menos el doble de la señal a trabajar(nyquist)

Fn = Fs/2;% Nyquist Frequency (Hz)

Wp = 450/Fn; % Passband Frequency Vector (Normalised)

Ws = 20/Fn; % Stopband Frequency Vector (Normalised)

Rp = 1; % Passband Ripple (dB)

Rs = 50; % Stopband Attenuation (dB)

[n,Wp] = ellipord(Wp,Ws,Rp,Rs); % Calculate Filter Order

[z,p,k] = ellip(n,Rp,Rs,Wp,'high'); % Default Here Is A Lowpass Filter

[sos,g] = zp2sos(z,p,k); % Use Second-Order-Section Implementation For Stability

Xfilter1 = filtfilt(sos,g,sig_notch); % Filter Signal (Here: 'EMG')

signal_passband = Xfilter1;

fft_emg_filt=fft(Xfilter1);

N1 = length(fft_emg_filt);

fft_emg_filt_abs= abs(fft_emg_filt/N1);

signal_fft = fft_emg_filt_abs/max(fft_emg_filt_abs);

%%

% Dominio del Tiempo

% (1) Enhanced Mean absolute value

feature1=jEMAV(Xfilter1);

% (2) Enhanced Wavelength

feature2=jEWL(Xfilter1);

% (3) Mean absolute value

feature3=jMAV(Xfilter1);

% (4) Wavelength

feature4=jWL(Xfilter1);

% (5) Zeros Crossing

thres=0.01; feature5=jZC(Xfilter1,thres);

% (6) Slope Sign Change

thres=0.01; feature6=jSSC(Xfilter1,thres);

% (7) Root Mean Square

feature7=jRMS(Xfilter1);

% (8) Average Amplitude Change

```

feature8=jAAC(Xfilter1);
% (9) Difference Absolute Standard Deviation Value
feature9=jDASDV(Xfilter1);
% (10) Log Detector
feature10=jLD(Xfilter1);
% (11) Modified Mean Absolute Value
feature11=jMMAV(Xfilter1);
% (12) Modified Mean Absolute Value 2
feature12=jMMAV2(Xfilter1);
% (13) Myopulse Percentage Rate
thres=0.016; feature13=jMYOP(Xfilter1,thres);
% (14) Simple Square Integral
feature14=jSSI(Xfilter1);
% (15) Variance of EMG
feature15=jVAR(Xfilter1);
% (16) Willison Amplitude
thres=0.01; feature16=jWAMP(Xfilter1,thres);
% (17) Maximum Fractal length
feature17=jMFL(Xfilter1);

%Features dominio Frecuencia
% (18) Median Frequency FMD (mediana)
feature18 = medfreq(Xfilter1);
% (19)Mean Frequency FMN (media)
feature19=meanfreq(Xfilter1);
% (20)Peaks
%feature20=findpeaks(Xfilter1);

% Feature vector
feat=[feature1,feature2,feature3,feature4,feature5,feature6,feature7,feature8,feature9,feature10,feature11,feature12,feature13,feature14,feature15,feature16,feature17,feature18,feature19];
end

```

Obtención de señales en archivos con extensión .txt

GetSignalFromFile.m

```
function [t,x] = GetSignalFromFile(filename)
```

```
f = fopen(filename,'r');
tx = fscanf(f, '%f %f', [2,inf]);
%t = transpose(tx(1,:));
%x = transpose(tx(2,:));
t = tx(1,:);
x = tx(2,:);
fclose(f);
end
```

Normalización de la señal

enbw.m

```
function bw = enbw(window, fs)
%ENBW Equivalent noise bandwidth
% BW = ENBW(WINDOW) returns the two-sided equivalent noise bandwidth for
% a uniformly sampled window whose coefficients are specified in the
% vector WINDOW. This bandwidth is normalized to the noise power per
% frequency bin.
%
% BW = ENBW(WINDOW, Fs) returns the two-sided equivalent noise bandwidth
% (in Hz) for a uniformly sampled window whose coefficients are specified
% in the vector WINDOW, where Fs is the sampling rate of the window.
%
% % Example 1:
% % Compute the equivalent noise bandwidth of a Hann window
% bw1 = enbw(hann(10000))
%
% % Example 2:
% % Compute the equivalent noise bandwidth (in Hz) of a Hann window
% % sampled at 44.1 kHz.
```

```

% bw2 = enbw(hann(10000), 44.1e3)

% Reference:
% [1] fredric j. harris [sic], On the Use of Windows for Harmonic
% Analysis with the Discrete Fourier Transform, Proceedings of
% the IEEE, Vol. 66, No. 1, January 1978. Eqn 11, 15.

% Copyright 2012-2013 The MathWorks, Inc.

% two-sided ENBW computation of a window.
validateattributes(window,{'numeric'},{'real','vector'}, ...
    'enbw','WINDOW',1);

% compute normalized ENBW
bw = (rms(window)/mean(window))^2;

% if Fs is specified, scale by the frequency bin width
if nargin > 1
    validateattributes(fs,{'numeric'},{'real','positive','scalar'}, ...
        'enbw','Fs',2);
    bw = bw * double(fs) / length(window);
end

```

Cálculo del RMS

rms.m

```
%% DECLARATIONS AND INITIALIZATIONS
```

```

% Calculates windowed (over- and non-overlapping) RMS of a signal using the specified
windowlength
% y = rms(signal, windowlength, overlap, zeropad)
% signal is a 1-D vector
% windowlength is an integer length of the RMS window in samples

```

% overlap is the number of samples to overlap adjacent windows (enter 0 to use non-overlapping windows)

% zeropad is a flag for zero padding the end of your data...(0 for NO, 1 for YES)

% ex. y=rms(mysignal, 30, 10, 1). Calculate RMS with window of length 30 samples, overlapped by 10 samples each, and zeropad the last window if necessary

% ex. y=rms(mysignal, 30, 0, 0). Calculate RMS with window of length 30 samples, no overlapping samples, and do not zeropad the last window

%

% Author: A. Bolu Ajiboye

```
function y = rms(signal, windowlength, overlap, zeropad)
```

```
delta = windowlength - overlap;
```

```
%% CALCULATE RMS
```

```
indices = 1:delta:length(signal);
```

```
% Zeropad signal
```

```
if length(signal) - indices(end) + 1 < windowlength
```

```
    if zeropad
```

```
        signal(end+1:indices(end)+windowlength-1) = 0;
```

```
    else
```

```
        indices = indices(1:find(indices+windowlength-1 <= length(signal), 1, 'last'));
    end
```

```
end
```

```
end
```

```
y = zeros(1, length(indices));
```

```
% Square the samples
```

```
signal = signal.^2;
```

```
index = 0;
```

```
for i = indices
```

```
    index = index+1;
```

```
    % Average and take the square root of each window
```

```
    y(index) = sqrt(mean(signal(i:i+windowlength-1)));
```

```
end
```

Cálculo de la frecuencia mediana

medfreq.m

```
function [f,pwr] = medfreq(varargin)
%MEDFREQ Median Frequency
%  FREQ = MEDFREQ(X) computes the median normalized angular frequency,
%  FREQ, of the power spectrum of the time-domain signal in vector X.
%  FREQ has units of radians/seconds.  If X is a matrix, MEDFREQ computes
%  the median frequency of each column in X independently.  MEDFREQ uses a
%  rectangular window when computing the spectrum.
%
%  The median frequency is defined as the frequency at which the power
%  spectrum is divided into two equal areas via rectangular integral
%  approximation.
%
%  FREQ = MEDFREQ(X, Fs) computes the median frequency, FREQ, of the power
%  spectrum of the time-domain signal in vector X with sample rate, Fs.
%  FREQ and Fs have units of hertz.
%
%  FREQ = MEDFREQ(Pxx, F) computes the median frequency of the power
%  spectral density estimate, Pxx.  F is a vector containing the
%  frequencies that correspond to the estimates given in Pxx and must
%  contain at least two elements.
%
%  FREQ = MEDFREQ(Sxx, F, RBW) computes the median frequency of the power
%  spectrum estimate, Sxx, with resolution bandwidth RBW.
%
%  FREQ = MEDFREQ(..., FREQRANGE) specifies FREQRANGE as a two-element
%  vector of real values, specifying the two frequencies between which you
%  want to compute the median frequency.  The default value for FREQRANGE
%  is the entire bandwidth of the input signal.
%
%  [FREQ,PWR] = MEDFREQ(...) also returns the bandpower, POWER, of the
%  spectrum.  If FREQRANGE is specified, then POWER will contain the
%  bandpower within the frequency range.
%
```

```

% MEDFREQ(...) with no output arguments will plot the PSD (or power
% spectrum) and annotate the median frequency.
%
% % Example 1:
% % Compute the median frequency of a chirp signal
%
% nSamp = 1024;
% Fs = 1024e3;
% t = (0:nSamp-1)/Fs;
% x = chirp(t,50e3,nSamp/Fs,100e3);
%
% medfreq(x,Fs)
%
% % Example 2:
% % Compute the median frequency of a sinusoid from a PSD estimate
%
% nSamp = 1024;
% Fs = 1024e3;
% t = (0:nSamp-1)/Fs;
% x = sin(2*pi*t*100.123e3);
%
% [Pxx, F] = periodogram(x,kaiser(nSamp,38),[],Fs);
% medfreq(Pxx,F)
%
% See also MEANFREQ BANDPOWER FINDPEAKS PERIODOGRAM PWELCH PLOMB

% Copyright 2014 The MathWorks, Inc.
marginchk(1,4);

% use a rectangular window for time-domain input
kaiserBeta = 0;

% fetch the PSD from the input
[Pxx, F, Frange, rbw, extraArgs, status] = psdparserange('medfreq', kaiserBeta, varargin{:});

% use full range if unspecified
if isempty(Frange)

```



```

    Frange = [F(1) F(end)];
end

% ensure no additional arguments are specified
if ~isempty(extraArgs)
    error(message('signal:medfreq:ExtraArgs'));
end

% compute the median frequency and power within the specified range
[f,pwr] = computeMedFreq(Pxx, F, Frange);

% plot if no output arguments specified
if nargin==0
    plotMedFreq(Pxx, F, Frange, rbw, f, status);
end

%-----
function [f,pwr] = computeMedFreq(Pxx, F, freqrange)

% Compute the power from the PSD
width = specfreqwidth(F);
P = bsxfun(@times,width,Pxx);

% Cumulative rectangular integration
cumPwr = [zeros(1,size(P,2)); cumsum(P)];

% place borders halfway between each estimate.
cumF = [F(1); (F(1:end-1)+F(2:end))/2; F(end)];

% find the integrated power for the low and high frequency range
Plo = interpPower(cumPwr,cumF,freqrange(1));
Phi = interpPower(cumPwr,cumF,freqrange(2));

% return the power between the frequency range
pwr = Phi-Plo;

```

% return the frequency that divides the power equally

```
f = interpFreq(cumPwr,cumF,(Plo+Phi)/2);
```

```
%-----
```

```
function p = interpPower(cumPwr,cumF,f)
```

```
idx = find(f<=cumF, 1,'first');
```

```
if ~isempty(idx)
```

```
    if idx==1
```

```
        p = signal.internal.linterp(cumPwr(1,:),cumPwr(2,:),cumF(1),cumF(2),f);
```

```
    else
```

```
        p = signal.internal.linterp(cumPwr(idx,:),cumPwr(idx-1,:), ...
            cumF(idx),cumF(idx-1),f);
```

```
    end
```

```
else
```

```
    p = nan(1,size(cumPwr,2));
```

```
end
```

```
%-----
```

```
function f = interpFreq(cumPwr, cumF, pwrThresh)
```

```
nChan = size(cumPwr,2);
```

```
f = zeros(1,nChan);
```

```
for iChan=1:nChan
```

```
    idx = find(pwrThresh(iChan)<=cumPwr(:,iChan),1,'first');
```

```
    if ~isempty(idx)
```

```
        if idx==1
```

```
            idx=2;
```

```
        end
```

```
        f(iChan) = signal.internal.linterp(cumF(idx-1), cumF(idx), ...
            cumPwr(idx-1,iChan), cumPwr(idx,iChan), pwrThresh(iChan));
```

```
    else
```

```
        f(iChan) = NaN;
```

```
    end
```

```
end
```

```
%-----
```

```
function plotMedFreq(Pxx, F, Frange, rbw, Fmed, status)
```

```

% plot spectrum
if strcmp(status.inputType,'power');
    % power spectrum when specified
    [hLine, xscale] = psdplot(Pxx, F, rbw, 'power', status);
else
    % otherwise, default to PSD
    [hLine, xscale] = psdplot(Pxx, F, rbw, 'psd', status);
end

% show the active frequency range of the measurement
hAxes = ancestor(hLine(1),'axes');
xLim = [F(1) F(end)];
yLim = get(hAxes,'YLim');
psdmaskactiverange(hAxes, xscale, xLim, yLim, Frange);

% plot vertical bar for each estimate
for i=1:numel(Fmed)
    line(xscale*[Fmed(i) Fmed(i)], yLim, ...
        'Parent',hAxes, ...
        'LineStyle','-.', ...
        'Color',get(hLine(i),'Color'));
end

% title the plot
titleStr = getString(message('signal:medfreq:MedianFreqEstimate'));
if isscalar(Fmed)
    [Fm, ~, units] = engunits(Fmed(1), 'unicode');
    if status.normF
        titleStr = sprintf('%s: %.3f \times \pi %srad/sample',titleStr,Fm/pi,units);
    else
        titleStr = sprintf('%s: %.3f %sHz',titleStr,Fm,units);
    end
end
end

title(titleStr);

```

Extracción de características

ProcessAllSignals.m

```

fd = fopen('filelist.txt','r');
if(fd < 0)
    error("No fue posible abrir la lista de archivos");
end

file_count = 0;

result_filename = 'emg result\feats.csv';
fd_out = fopen(result_filename, 'w');
fprintf(fd_out, 'Signal Name, Enhanced Mean absolute value, Enhanced Wavelength, Mean
absolute value,Wavelength,Zeros Crossing,Slope Sign Change,Root Mean Square, Average
Amplitude Change,Difference Absolute Standard Deviation Value,Log Detector,Modified Mean
Absolute Value, Modified Mean Absolute Value 2,Myopulse Percentage Rate,Simple Square
Integral,Variance of EMG,Willison Amplitude,Maximum Fractal length,Median Frequency,Mean
Frequency\r\n');

while ~feof(fd)
    filename = fgetl(fd);
    file_count = file_count + 1;

    [feat, signal, signal_notch, signal_passband, signal_fft] = Process_Signal(filename);

    [path,name,extension] = fileparts(filename);

    fprintf(fd_out, '%s,', name);
    for n = 1:max(size(feat))
        fprintf(fd_out, '%f,', feat(n));
    end
    fprintf(fd_out, '\r\n');

    if exist(filename, 'file') == 0

```

```

    display(filename);
    error("Archivo no encontrado");
end
display(file_count);
end
fclose(fd_out);
fclose(fd);
display("End of file processing");

```

Filtrado

Filtros.m

```
EMG = X1;
```

```
% Sampling Frequency (Hz)
```

```
Fs = 1000;% Frecuencia de muestreo tiene que ser por lo menos el doble de la señal a
trabajar(nyquist)
```

```
Fn = Fs/2;% Nyquist Frequency (Hz)
```

```
Wp = 450/Fn; % Passband Frequency Vector (Normalised)
```

```
Ws = 20/Fn; % Stopband Frequency Vector (Normalised)
```

```
Rp = 1; % Passband Ripple (dB)
```

```
Rs = 50; % Stopband Attenuation (dB)
```

```
[n,Wp] = ellipord(Wp,Ws,Rp,Rs); % Calculate Filter Order
```

```
[z,p,k] = ellip(n,Rp,Rs,Wp,'high'); % Default Here Is A Lowpass Filter
```

```
[sos,g] = zp2sos(z,p,k); % Use Second-Order-Section Implementation For
```

```
Stability
```

```
EMG_filtered = filtfilt(sos,g,EMG); % Filter
```

```
figure
```

```
plot(t1, [EMG_filtered, EMG]);
```

```
fft_emg=fft(EMG);
```

```
fft_emgflt=fft(EMG_filtered);
```

```

N1 = length(fft_emg_ft);

fft_emg_abs= abs(fft_emg/N1);
fft_emg_ft_abs= abs(fft_emg_ft/N1);

fft_emg_abs = fft_emg_abs/max(fft_emg_abs);
fft_emg_ft_abs = fft_emg_ft_abs/max(fft_emg_ft_abs);

F1axis = (0:N1/2)*fs/N1;
figure;
plot(F1axis, [fft_emg_abs(1:(N1/2+1)), fft_emg_ft_abs(1:(N1/2+1))]); xlabel('Número de muestras');
ylabel('Voltaje (mV)'); grid on;
title('FFT de la señal');

```

Filtro Pasa Banda

PasaBanda.m

```

function Hd = PasaBanda
%Ejemplo para utilizar la funcion:
%FP=PasaBanda; %se almacena el filtro en una variable FP
%fvtool(FP) %para visualizar el filtro
%OutFiltPass=filter(FP,y); % aplicar el filtro FP a una señal de entrada Y
%%plot(OutFiltPass) % para ver la señal filtrada

% Equiripple Bandpass filter designed using the FIRPM function.

% All frequency values are in Hz.
Fs = 1000; % Sampling Frequency

Fstop1 = 4; % First Stopband Frequency
Fpass1 = 7; % First Passband Frequency
Fpass2 = 400; % Second Passband Frequency
Fstop2 = 410; % Second Stopband Frequency
Dstop1 = 0.0001; % First Stopband Attenuation
Dpass = 0.057501127785; % Passband Ripple
Dstop2 = 0.0001; % Second Stopband Attenuation

```

```

dens = 20;          % Density Factor

% calcular el orden de los parámetros usando FIRPMORD.
[N, Fo, Ao, W] = firpmord([Fstop1 Fpass1 Fpass2 Fstop2]/(Fs/2), [0 1 ...
                        0], [Dstop1 Dpass Dstop2]);

% calcular los coeficientes usando la función FIRPM
b = firpm(N, Fo, Ao, W, {dens});
Hd = dfilt.dffir(b);

% [EOF]

```

Graficar señales

Graficar.m

```

%1000 muestras por segundo, resolución de 14 bits

filename1= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Normal\UCI Machine Learning
Repository\N_TXT\SN01-UCI_Reposo_Biceps_Femoral-nofilter.txt';
[t1,X1] = GetSignalFromFile(filename1);
figure ('Name','Sano');
plot(t1,X1);
title('P01 Reposo Biceps Femoral Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');

filename2= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Norma\UCI Machine Learning
Repository\N_TXT\SN01-UCI_Reposo_EMG_Semitendensioso-nofilter.txt';
[t2,X2] = GetSignalFromFile(filename2);
figure ('Name','Sano');
plot(t2,X2);
title('P01 Reposo Semitendinoso Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');

```

```
filename3= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Normal\UCI Machine Learning  
Repository\N_TXT\SN01-UCI_Reposo_Recto_Femoral-nofilter.txt';  
[t3,X3] = GetSignalFromFile(filename3);  
figure ('Name','Sano');  
plot(t3,X3);  
title('P01 Reposo Recto Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');
```

```
filename4= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Normal\UCI Machine Learning  
Repository\N_TXT\SN01-UCI_Reposo_Vasto_Medial-nofilter.txt';  
[t4,X4] = GetSignalFromFile(filename4);  
figure ('Name','Sano');  
plot(t4,X4);  
title('P01 Reposo Vasto Medial Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%SN01-UCI_Reposo_Vasto_Medial-nofilter
```

```
filename5= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Normal\UCI Machine Learning  
Repository\N_TXT\SN02-UCI_Reposo_Biceps_Femoral-nofilter.txt';  
[t5,X5] = GetSignalFromFile(filename5);  
figure ('Name','Sano');  
plot(t5,X5);  
title('P01 Reposo Biceps Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%SN02-UCI_Reposo_Biceps_Femoral-nofilter
```

```
filename6= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Normal\UCI Machine Learning  
Repository\N_TXT\SN02-UCI_Reposo_EMG_Semitendensioso-nofilter.txt';  
[t6,X6] = GetSignalFromFile(filename6);  
figure ('Name','Sano');  
plot(t6,X6);  
title('P02 Reposo Semitendensioso Sin Filtros SEMG');  
xlabel('Tiempo');
```



```
ylabel('Voltaje mV');
%SN02-UCI_Reposo_EMG_Semitendensioso-nofilter
```

```
filename7= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Normal\UCI Machine Learning
Repository\N_TXT\SN02-UCI_Reposo_Recto_Femoral-nofilter.txt';
[t7,X7] = GetSignalFromFile(filename7);
figure ('Name','Sano');
plot(t7,X7);
title('P02 Reposo Recto Femoral Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');
%SN02-UCI_Reposo_Recto_Femoral-nofilter
```

```
filename8= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Normal\UCI Machine Learning
Repository\N_TXT\SN02-UCI_Reposo_Vasto_Medial-nofilter.txt';
[t8,X8] = GetSignalFromFile(filename8);
figure ('Name','Sano');
plot(t8,X8);
title('P02 Reposo Vasto Medial Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');
%SN02-UCI_Reposo_Vasto_Medial-nofilter
```

```
filename9= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Normal\UCI Machine Learning
Repository\N_TXT\SN06-UCI_Reposo_Biceps_Femoral-nofilter.txt';
[t9,X9] = GetSignalFromFile(filename9);
figure ('Name','Sano');
plot(t9,X9);
title('P06 Reposo Biceps Femoral Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');
%SN06-UCI_Reposo_Biceps_Femoral-nofilter
```

```
filename10= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Normal\UCI Machine Learning
Repository\N_TXT\SN06-UCI_Reposo_EMG_Semitendensioso-nofilter.txt';
[t10,X10] = GetSignalFromFile(filename10);
figure ('Name','Sano');
```

```
plot(t10,X10);  
title('P06 Reposo Semitendensioso Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%SN06-UCI_Reposo_EMG_Semitendensioso-nofilter
```

```
filename11= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Normal\UCI Machine Learning  
Repository\N_TXT\SN06-UCI_Reposo_Recto_Femoral-nofilter.txt';  
[t11,X11] = GetSignalFromFile(filename11);  
figure ('Name','Sano');  
plot(t11,X11);  
title('P06 Reposo Recto Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%SN06-UCI_Reposo_Recto_Femoral-nofilter
```

```
filename12= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Normal\UCI Machine Learning  
Repository\N_TXT\SN06-UCI_Reposo_Vasto_Medial-nofilter.txt';  
[t12,X12] = GetSignalFromFile(filename12);  
figure ('Name','Sano');  
plot(t12,X12);  
title('P06 Reposo Vasto Medial Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%SN06-UCI_Reposo_Vasto_Medial-nofilter
```

```
filename13= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S01-UCI_Reposo_Biceps_Femoral-nofilter.txt';  
[t13,X13] = GetSignalFromFile(filename13);  
figure ('Name','MN');  
plot(t13,X13);  
title('P01 Reposo Biceps_Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S01-UCI_Reposo_Biceps_Femoral-nofilter
```

```
filename14= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S01-UCI_Reposo_EMG_Semitendensioso-nofilter.txt';  
[t14,X14] = GetSignalFromFile(filename14);  
figure ('Name','MN');  
plot(t14,X14);  
title('P01 Reposo Semitendensioso Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S01-UCI_Reposo_EMG_Semitendensioso-nofilter
```

```
filename15= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S01-UCI_Reposo_Recto_Femoral-nofilter.txt';  
[t15,X15] = GetSignalFromFile(filename15);  
figure ('Name','MN');  
plot(t15,X15);  
title('P01 Reposo Recto Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S01-UCI_Reposo_Recto_Femoral-nofilter
```

```
filename16= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S01-UCI_Reposo_Vasto_Medial-nofilter.txt';  
[t16,X16] = GetSignalFromFile(filename16);  
figure ('Name','MN');  
plot(t16,X16);  
title('P01 Reposo Vasto Medial Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S01-UCI_Reposo_Vasto_Medial-nofilter
```

```
filename17= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S02-UCI_Reposo_Biceps_Femoral-nofilter.txt';  
[t17,X17] = GetSignalFromFile(filename17);  
figure ('Name','MN');  
plot(t17,X17);  
title('P02 Reposo Biceps Femoral Sin Filtros SEMG');  
xlabel('Tiempo');
```

```
ylabel('Voltaje mV');  
%S02-UCI_Reposo_Biceps_Femoral-nofilter
```

```
filename18= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S02-UCI_Reposo_EMG_Semitendensioso-nofilter.txt';  
[t18,X18] = GetSignalFromFile(filename18);  
figure ('Name','MN');  
plot(t18,X18);  
title('P02 Reposo Semitendensioso Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S02-UCI_Reposo_EMG_Semitendensioso-nofilter
```

```
filename19= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S02-UCI_Reposo_Recto_Femoral-nofilter.txt';  
[t19,X19] = GetSignalFromFile(filename19);  
figure ('Name','MN');  
plot(t19,X19);  
title('P02 Reposo Recto Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S02-UCI_Reposo_Recto_Femoral-nofilter
```

```
filename20= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S02-UCI_Reposo_Vasto_Medial-nofilter.txt';  
[t20,X20] = GetSignalFromFile(filename20);  
figure ('Name','MN');  
plot(t20,X20);  
title('P02 Reposo Vasto Medial Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S02-UCI_Reposo_Vasto_Medial-nofilter
```

```
filename21= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S03-UCI_Reposo_Biceps_Femoral-nofilter.txt';  
[t21,X21] = GetSignalFromFile(filename21);  
figure ('Name','MN');
```

```
plot(t21,X21);  
title('P03 Reposo Biceps Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S03-UCI_Reposo_Biceps_Femoral-nofilter
```

```
filename22= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S03-UCI_Reposo_EMG_Semitendensioso-nofilter.txt';  
[t22,X22] = GetSignalFromFile(filename22);  
figure ('Name','MN');  
plot(t22,X22);  
title('P03 Reposo Semitendensioso Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S03-UCI_Reposo_EMG_Semitendensioso-nofilter
```

```
filename23= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S03-UCI_Reposo_Recto_Femoral-nofilter.txt';  
[t23,X23] = GetSignalFromFile(filename23);  
figure ('Name','MN');  
plot(t23,X23);  
title('P03 Reposo Recto Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S03-UCI_Reposo_Recto_Femoral-nofilter
```

```
filename24= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S03-UCI_Reposo_Vasto_Medial-nofilter.txt';  
[t24,X24] = GetSignalFromFile(filename24);  
figure ('Name','MN');  
plot(t24,X24);  
title('P03 Reposo Vasto Medial Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S03-UCI_Reposo_Vasto_Medial-nofilter
```

```
filename25= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning
Repository - no classification\A_TXT\S04-UCI_Reposo_Biceps_Femoral-nofilter.txt';
[t25,X25] = GetSignalFromFile(filename25);
figure ('Name','MN');
plot(t25,X25);
title('P04 Reposo Biceps Femoral Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');
%S04-UCI_Reposo_Biceps_Femoral-nofilter
```

```
filename26= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning
Repository - no classification\A_TXT\S04-UCI_Reposo_EMG_Semitendensioso-nofilter.txt';
[t26,X26] = GetSignalFromFile(filename26);
figure ('Name','MN');
plot(t26,X26);
title('P04 Reposo Semitendensioso Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');
%S04-UCI_Reposo_EMG_Semitendensioso-nofilter
```

```
filename27= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning
Repository - no classification\A_TXT\S04-UCI_Reposo_Recto_Femoral-nofilter.txt';
[t27,X27] = GetSignalFromFile(filename27);
figure ('Name','MN');
plot(t27,X27);
title('P04 Reposo Recto Femoral Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');
%S04-UCI_Reposo_Recto_Femoral-nofilter
```

```
filename28= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning
Repository - no classification\A_TXT\S04-UCI_Reposo_Vasto_Medial-nofilter.txt';
[t28,X28] = GetSignalFromFile(filename28);
figure ('Name','MN');
plot(t28,X28);
title('P04 Reposo Vasto Medial Sin Filtros SEMG');
xlabel('Tiempo');
```

```
ylabel('Voltaje mV');
%S04-UCI_Reposo_Vasto_Medial-nofilter
```

```
filename29= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning
Repository - no classification\A_TXT\S05-UCI_Reposo_Biceps_Femoral-nofilter.txt';
[t29,X29] = GetSignalFromFile(filename29);
figure ('Name','MN');
plot(t29,X29);
title('P05 Reposo Biceps Femoral Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');
%S05-UCI_Reposo_Biceps_Femoral-nofilter
```

```
filename30= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning
Repository - no classification\A_TXT\S05-UCI_Reposo_EMG_Semitendensioso-nofilter.txt';
[t30,X30] = GetSignalFromFile(filename30);
figure ('Name','MN');
plot(t30,X30);
title('P05 Reposo Semitendensioso Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');
%S05-UCI_Reposo_EMG_Semitendensioso-nofilter
```

```
filename31= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning
Repository - no classification\A_TXT\S05-UCI_Reposo_Recto_Femoral-nofilter.txt';
[t31,X31] = GetSignalFromFile(filename31);
figure ('Name','MN');
plot(t31,X31);
title('P05 Reposo Recto Femoral Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');
%S05-UCI_Reposo_Recto_Femoral-nofilter
```

```
filename32= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning
Repository - no classification\A_TXT\S05-UCI_Reposo_Vasto_Medial-nofilter.txt';
[t32,X32] = GetSignalFromFile(filename32);
figure ('Name','MN');
```

```
plot(t32,X32);  
title('P05 Reposo Vasto Medial Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S05-UCI_Reposo_Vasto_Medial-nofilter
```

```
filename33= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S06-UCI_Reposo_Biceps_Femoral-nofilter.txt';  
[t33,X33] = GetSignalFromFile(filename33);  
figure ('Name','MN');  
plot(t33,X33);  
title('P06 Reposo Biceps Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S06-UCI_Reposo_Biceps_Femoral-nofilter
```

```
filename34= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S06-UCI_Reposo_EMG_Semitendensioso-nofilter.txt';  
[t34,X34] = GetSignalFromFile(filename34);  
figure ('Name','MN');  
plot(t34,X34);  
title('P06 Reposo Semitendensioso Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S06-UCI_Reposo_EMG_Semitendensioso-nofilter
```

```
filename35= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S06-UCI_Reposo_Recto_Femoral-nofilter.txt';  
[t35,X35] = GetSignalFromFile(filename35);  
figure ('Name','MN');  
plot(t35,X35);  
title('P06 Reposo Recto Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S06-UCI_Reposo_Recto_Femoral-nofilter
```



```
filename36= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S06-UCI_Reposo_Vasto_Medial-nofilter.txt';  
[t36,X36] = GetSignalFromFile(filename36);  
figure ('Name','MN');  
plot(t36,X36);  
title('P06 Reposo Vasto Medial Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S06-UCI_Reposo_Vasto_Medial-nofilter
```

```
filename37= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S07-UCI_Reposo_Biceps_Femoral-nofilter.txt';  
[t37,X37] = GetSignalFromFile(filename37);  
figure ('Name','MN');  
plot(t37,X37);  
title('P07 Reposo Biceps Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S07-UCI_Reposo_Biceps_Femoral-nofilter
```

```
filename38= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S07-UCI_Reposo_EMG_Semitendensioso-nofilter.txt';  
[t38,X38] = GetSignalFromFile(filename38);  
figure ('Name','MN');  
plot(t38,X38);  
title('P07 Reposo Semitendensioso Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S07-UCI_Reposo_EMG_Semitendensioso-nofilter
```

```
filename39= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S07-UCI_Reposo_Recto_Femoral-nofilter.txt';  
[t39,X39] = GetSignalFromFile(filename39);  
figure ('Name','MN');  
plot(t39,X39);  
title('P07 Reposo Recto Femoral Sin Filtros SEMG');  
xlabel('Tiempo');
```

```
ylabel('Voltaje mV');  
%S07-UCI_Reposo_Recto_Femoral-nofilter
```

```
filename40= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S07-UCI_Reposo_Vasto_Medial-nofilter.txt';  
[t40,X40] = GetSignalFromFile(filename40);  
figure ('Name','MN');  
plot(t40,X40);  
title('P07 Reposo Vasto Medial Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S07-UCI_Reposo_Vasto_Medial-nofilter
```

```
filename41= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S08-UCI_Reposo_Biceps_Femoral-nofilter.txt';  
[t41,X41] = GetSignalFromFile(filename41);  
figure ('Name','MN');  
plot(t41,X41);  
title('P08 Reposo Biceps Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S08-UCI_Reposo_Biceps_Femoral-nofilter
```

```
filename42= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S08-UCI_Reposo_EMG_Semitendensioso-nofilter.txt';  
[t42,X42] = GetSignalFromFile(filename42);  
figure ('Name','MN');  
plot(t42,X42);  
title('P08 Reposo Semitendensioso Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S08-UCI_Reposo_EMG_Semitendensioso-nofilter
```

```
filename43= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S08-UCI_Reposo_Recto_Femoral-nofilter.txt';  
[t43,X43] = GetSignalFromFile(filename43);  
figure ('Name','MN');
```

```
plot(t43,X43);  
title('P08 Reposo Recto Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S08-UCI_Reposo_Recto_Femoral-nofilter
```

```
filename44= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S08-UCI_Reposo_Vasto_Medial-nofilter.txt';  
[t44,X44] = GetSignalFromFile(filename44);  
figure ('Name','MN');  
plot(t44,X44);  
title('P08 Reposo Vasto_Medial Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S08-UCI_Reposo_Vasto_Medial-nofilter
```

```
filename45= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S09-UCI_Reposo_Biceps_Femoral-nofilter.txt';  
[t45,X45] = GetSignalFromFile(filename45);  
figure ('Name','MN');  
plot(t45,X45);  
title('P09 Reposo Biceps Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S09-UCI_Reposo_Biceps_Femoral-nofilter
```

```
filename46= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S09-UCI_Reposo_EMG_Semitendensioso-nofilter.txt';  
[t46,X46] = GetSignalFromFile(filename46);  
figure ('Name','MN');  
plot(t46,X46);  
title('P09 Reposo Semitendensioso Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S09-UCI_Reposo_EMG_Semitendensioso-nofilter
```

```
filename47= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S09-UCI_Reposo_Recto_Femoral-nofilter.txt';  
[t47,X47] = GetSignalFromFile(filename47);  
figure ('Name','MN');  
plot(t47,X47);  
title('P09 Reposo Recto Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S09-UCI_Reposo_Recto_Femoral-nofilter
```

```
filename48= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S09-UCI_Reposo_Vasto_Medial-nofilter.txt';  
[t48,X48] = GetSignalFromFile(filename48);  
figure ('Name','MN');  
plot(t48,X48);  
title('P09 Reposo Vasto Medial Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S09-UCI_Reposo_Vasto_Medial-nofilter
```

```
filename49= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S10-UCI_Reposo_Biceps_Femoral-nofilter.txt';  
[t49,X49] = GetSignalFromFile(filename49);  
figure ('Name','MN');  
plot(t49,X49);  
title('P10 Reposo Biceps Femoral Sin Filtros SEMG');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%S10-UCI_Reposo_Biceps_Femoral-nofilter
```

```
filename50= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning  
Repository - no classification\A_TXT\S10-UCI_Reposo_EMG_Semitendensioso-nofilter.txt';  
[t50,X50] = GetSignalFromFile(filename50);  
figure ('Name','MN');  
plot(t50,X50);  
title('P10 Reposo Semitendensioso Sin Filtros SEMG');  
xlabel('Tiempo');
```

```
ylabel('Voltaje mV');
%S10-UCI_Reposo_EMG_Semitendensioso-nofilter

filename51= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning
Repository - no classification\A_TXT\S10-UCI_Reposo_Recto_Femoral-nofilter.txt';
[t51,X51] = GetSignalFromFile(filename51);
figure ('Name','MN');
plot(t51,X51);
title('P10 Reposo Recto_Femoral Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');
%S10-UCI_Reposo_Recto_Femoral-nofilter

filename52= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning
Repository - no classification\A_TXT\S10-UCI_Reposo_Vasto_Medial-nofilter.txt';
[t52,X52] = GetSignalFromFile(filename52);
figure ('Name','MN');
plot(t52,X52);
title('P10 Reposo Vasto Medial Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');
%S10-UCI_Reposo_Vasto_Medial-nofilter

filename53= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning
Repository - no classification\A_TXT\S11-UCI_Reposo_Biceps_Femoral-nofilter.txt';
[t53,X53] = GetSignalFromFile(filename53);
figure ('Name','MN');
plot(t53,X53);
title('P11 Reposo Biceps Femoral Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');
%S11-UCI_Reposo_Biceps_Femoral-nofilter

filename54= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning
Repository - no classification\A_TXT\S11-UCI_Reposo_EMG_Semitendensioso-nofilter.txt';
[t54,X54] = GetSignalFromFile(filename54);
figure ('Name','MN');
```

```

plot(t54,X54);
title('P11 Reposo Semitendensioso Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');
%S11-UCI_Reposo_EMG_Semitendensioso-nofilter

```

```

filename55= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning
Repository - no classification\A_TXT\S11-UCI_Reposo_Recto_Femoral-nofilter.txt';
[t55,X55] = GetSignalFromFile(filename55);
figure ('Name','MN');
plot(t55,X55);
title('P11 Reposo Recto Femoral Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');
%S11-UCI_Reposo_Recto_Femoral-nofilter

```

```

filename56= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\UCI Machine Learning
Repository - no classification\A_TXT\S11-UCI_Reposo_Vasto_Medial-nofilter.txt';
[t56,X56] = GetSignalFromFile(filename56);
figure ('Name','MN');
plot(t56,X56);
title('P11 Reposo Vasto Medial Sin Filtros SEMG');
xlabel('Tiempo');
ylabel('Voltaje mV');
%S11-UCI_Reposo_Vasto_Medial-nofilter

```

```

%10000/Voltaje mV
filename57= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG
Normal\physionet\emg_healthy_S1.txt';
[t57,X57] = GetSignalFromFile(filename57);
figure ('Name','Sano');
plot(t57,X57);
title('P01 Reposo tibialis anterior muscle Filtrado Digital');
xlabel('Tiempo');
ylabel('Voltaje mV');
%emg_healthy_S1

```

```
filename58= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG
Normal\physionet\emg_healthy_S2.txt';
[t58,X58] = GetSignalFromFile(filename58);
figure ('Name','Sano');
plot(t58,X58);
title('P02 Reposo tibialis anterior muscle Filtrado Digital');
xlabel('Tiempo');
ylabel('Voltaje mV');
%emg_healthy_S2
```

```
filename59= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\EMG
Miopatía\physionet\emg_myopathy_S1.txt';
[t59,X59] = GetSignalFromFile(filename59);
figure ('Name','Miopatía');
plot(t59,X59);
title('P01 Reposo Tibialis Anterior Filtrado Digital');
xlabel('Tiempo');
ylabel('Voltaje mV');
%emg_myopathy_S1
```

```
filename60= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\EMG
Miopatía\physionet\emg_myopathy_S2.txt';
[t60,X60] = GetSignalFromFile(filename60);
figure ('Name','Miopatía');
plot(t60,X60);
title('P02 Reposo tibialis anterior muscle Filtrado Digital');
xlabel('Tiempo');
ylabel('Voltaje mV');
%emg_myopathy_S2
```

```
filename61= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\EMG
Miopatía\physionet\emg_myopathy_S3.txt';
[t61,X61] = GetSignalFromFile(filename61);
figure ('Name','Miopatía');
plot(t61,X61);
```

```
title('P03 Reposo tibialis anterior muscle Filtrado Digital');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%emg_myopathy_S3
```

```
filename62= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\EMG  
Miopatía\physionet\emg_myopathy_S4.txt';  
[t62,X62] = GetSignalFromFile(filename62);  
figure ('Name','Miopatía');  
plot(t62,X62);  
title('P04 Reposo tibialis anterior muscle Filtrado Digital');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%emg_myopathy_S4
```

```
filename63= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\EMG  
Miopatía\physionet\emg_myopathy_S5.txt';  
[t63,X63] = GetSignalFromFile(filename63);  
figure ('Name','Miopatía');  
plot(t63,X63);  
title('P05 Reposo tibialis anterior muscle Filtrado Digital');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%emg_myopathy_S5
```

%NEUROPATÍA

```
filename64= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\EMG  
Neuropatía\physionet\emg_neuropathy_S1.txt';  
[t64,X64] = GetSignalFromFile(filename64);  
figure ('Name','Neuropatía');  
plot(t64,X64);  
title('P01 Reposo tibialis anterior muscle Filtrado Digital');  
xlabel('Tiempo');  
ylabel('Voltaje mV');  
%emg_neuropathy_S1
```



```
filename65= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\EMG
Neuropatía\physionet\emg_neuropathy_S2.txt';
[t65,X65] = GetSignalFromFile(filename65);
figure ('Name','Neuropatía');
plot(t65,X65);
title('P02 Reposo tibialis anterior muscle Filtrado Digital');
xlabel('Tiempo');
ylabel('Voltaje mV');
%emg_neuropathy_S2
```

```
filename66= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\EMG
Neuropatía\physionet\emg_neuropathy_S3.txt';
[t66,X66] = GetSignalFromFile(filename66);
figure ('Name','Neuropatía');
plot(t66,X66);
title('P03 Reposo tibialis anterior muscle Filtrado Digital');
xlabel('Tiempo');
ylabel('Voltaje mV');
%emg_neuropathy_S3
```

```
filename67= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\EMG
Neuropatía\physionet\emg_neuropathy_S4.txt';
[t67,X67] = GetSignalFromFile(filename67);
figure ('Name','Neuropatía');
plot(t67,X67);
title('P04 Reposo tibialis anterior muscle Filtrado Digital');
xlabel('Tiempo');
ylabel('Voltaje mV');
%emg_neuropathy_S4
```

```
filename68= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\EMG
Neuropatía\physionet\emg_neuropathy_S5.txt';
[t68,X68] = GetSignalFromFile(filename68);
figure ('Name','Neuropatía');
plot(t68,X68);
title('P05 Reposo tibialis anterior muscle Filtrado Digital');
xlabel('Tiempo');
```

```
ylabel('Voltaje mV');
%emg_neuropathy_S5
```

```
filename69= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Anormal\EMG
Neuropatía\physionet\emg_neuropathy_S6.txt';
[t69,X69] = GetSignalFromFile(filename69);
figure ('Name','Neuropatía');
plot(t69,X69);
title('P06 Reposo tibialis anterior muscle Filtrado Digital');
xlabel('Tiempo');
ylabel('Voltaje mV');
%emg_neuropathy_S6
```

```
filename70= 'C:\Users\Cris\Desktop\BD Diagnóstico\Señales EMG Normal\emglab\C01\Biceps
Brachii Long Head\N2001C01BB51.bin';
[t70,X70] = GetSignalFromFile(filename70);
figure ('Name','Neuropatía');
plot(t70,X70);
title('P07 Reposo tibialis anterior muscle Filtrado Digital');
xlabel('Tiempo');
ylabel('Voltaje mV');
%emg_neuropathy_S7
```

Módulo RMS de la extracción de características

Getrmsfeat.m

```
%
% GETRMSFEAT Gets the RMS feature (Root Mean Square).
%
% feat = getrmsfeat(x,winsize,wininc,datawin,dispstatus)
%
% Author Adrian Chan
%
% This function computes the RMS feature of the signals in x,
% which are stored in columns.
```

```

%
% The signals in x are divided into multiple windows of size
% winsize and the windows are space wininc apart.
%
% Inputs
% x:      columns of signals
% winsize: window size (length of x)
% wininc:  spacing of the windows (winsize)
% datawin: window for data (e.g. Hamming, default rectangular)
%          must have dimensions of (winsize,1)
% dispstatus: zero for no waitbar (default)
%
% Outputs
% feat:   RMS value in a 2 dimensional matrix
%         dim1 window
%         dim2 feature (col i is the features for the signal in column i of x)
%
% Modifications
% 05/01/14 AC Change feat output so that dim1 is window and dim2 is feature
% 04/06/23 AC First created.

```

```

function feat = getrmsfeat(x,winsize,wininc,datawin,dispstatus)

```

```

if nargin < 5
    if nargin < 4
        if nargin < 3
            if nargin < 2
                winsize = size(x,1);
            end
            wininc = winsize;
        end
        datawin = ones(winsize,1);
    end
    dispstatus = 0;
end

```

```

datasize = size(x,1);

```

170

```
Nsignals = size(x,2);
numwin = floor((datasize - winsize)/wininc)+1;

% allocate memory
feat = zeros(numwin,Nsignals);

if dispstatus
    h = waitbar(0,'Computing RMS features...');
end

st = 1;
en = winsize;

for i = 1:numwin
    if dispstatus
        waitbar(i/numwin);
    end
    curwin = x(st:en,:).*repmat(datawin,1,Nsignals);
    feat(i,:) = sqrt(mean(curwin.^2));

    st = st + wininc;
    en = en + wininc;
end

if dispstatus
    close(h)
end
```

Anexo 4. Detalle parámetros de la red neuronal

Parámetros relacionados al entrenamiento

1. Épocas (epochs): es el número de veces que el conjunto de entrenamiento completo se muestra a la red neuronal mientras entrena.
2. Tamaño de lote (batch size): es el número que define cuántas muestras tiene el subconjunto que se alimenta a la red. Ésto se repite hasta terminar de pasar todo el conjunto de entrenamiento
3. Optimizador (optimizer): existe una cantidad de algoritmos distintos que optimizan el proceso en que los valores de los pesos y la parcialidad se van actualizando. Estos algoritmos buscan disminuir la función de error o maximizar la función objetivo. Usando valores del gradiente con respecto a los parámetros, la derivada indica si la función va incrementando o disminuyendo en cierto punto con una línea tangente en su superficie. El optimizador define la función de pérdida u objetivo a utilizar y su comportamiento depende de la tasa de aprendizaje
4. Tasa de aprendizaje (learning rate): La tasa de aprendizaje determina qué tan rápidamente las actualizaciones del gradiente siguen la dirección del gradiente. Si la velocidad de aprendizaje es muy pequeña, el modelo converge muy lentamente; si la tasa de aprendizaje es demasiado grande, el modelo diverge. Para las redes neuronales comunes se establece típicamente entre 0.1 y 0.00001. Este parámetro va altamente relacionado al tipo de optimizador que se utiliza. (MATLAB, 2004)

La siguiente lista especifica los algoritmos que se prueban y los acrónimos que usamos para identificarlos.

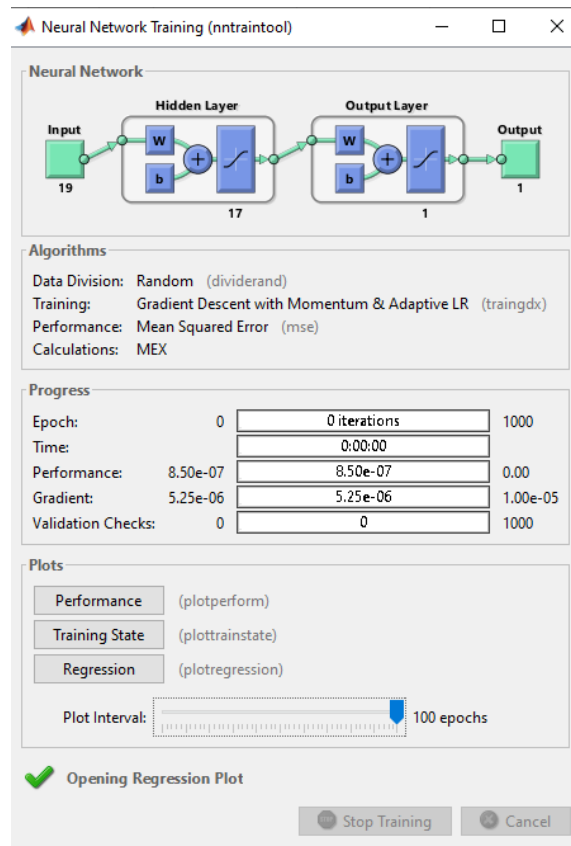
Acrónimo	Algoritmo
LM trainlm	Levenberg-Marquardt
BFG trainbfg	BFGS Quasi-Newton
RP trainrp	Resilient Backpropagation
SCG trainscg	Scaled Conjugate Gradient
CGB traincgb	Conjugate Gradient with Powell/Beale Restarts
CGF traincgf	Fletcher-Powell Conjugate Gradient
CGP traincgp	Polak-Ribière Conjugate Gradient
OSS trainoss	One-Step Secant

GDX traingdx Variable Learning Rate Backpropagation

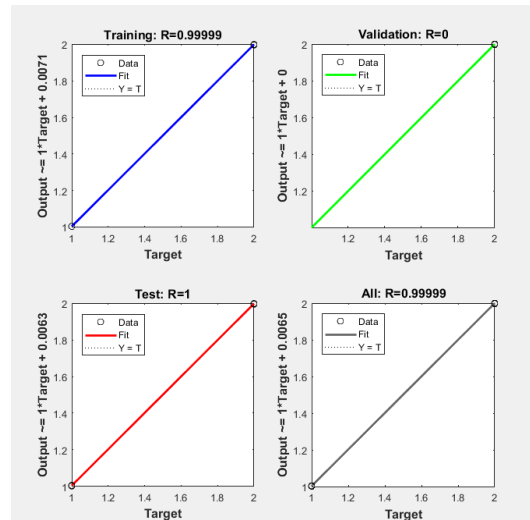
Al igual que los métodos cuasi-Newton, el algoritmo de Levenberg-Marquardt f también conocido como el método de mínimos cuadrados amortiguados (DLS), se utiliza para resolver problemas de mínimos cuadrados no lineales. Estos problemas de minimización surgen especialmente en el ajuste de curvas de mínimos cuadrados. El detalle del algoritmo se encuentra en la página 5-28 del manual del toolbox de redes neuronales. (MATLAB, 2004)

Anexo 5. Resultados más significativos de los experimentos

Prueba 87

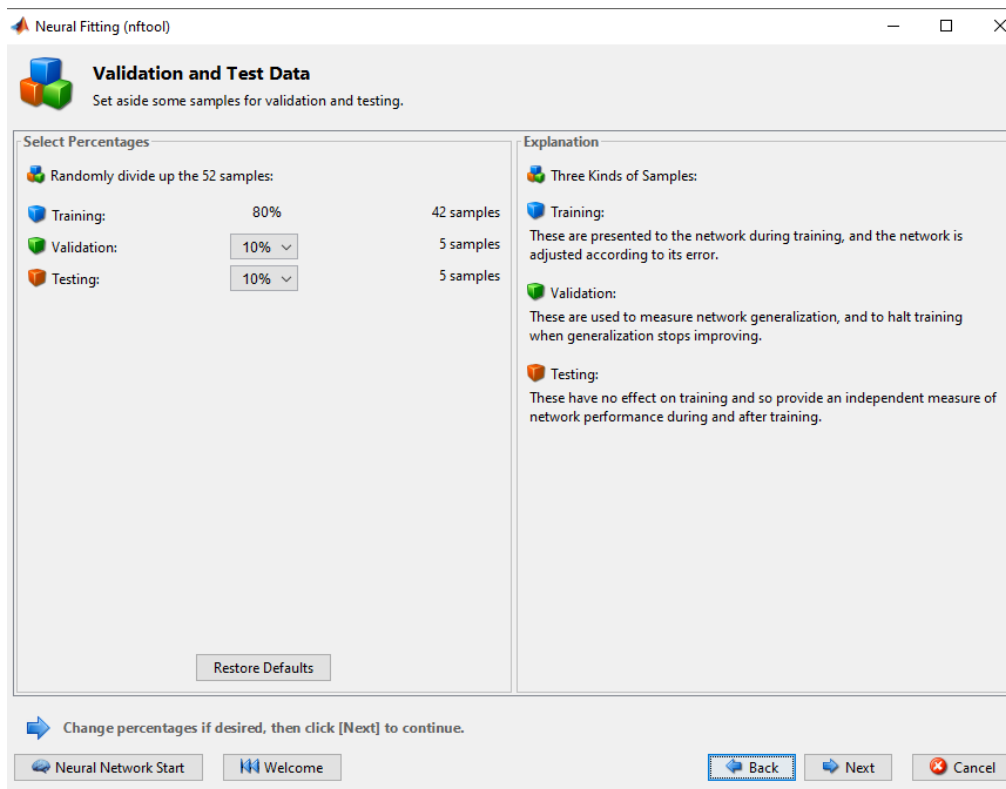


Arquitectura de la red neuronal prueba 87



Gráfica de regresión prueba 87

Pruebas 80% entrenamiento, 10% validación, 10% pruebas con 52 muestras



Neural Fitting (nftool)

Train Network

Train the network to fit the inputs and targets.

Train Network

Choose a training algorithm:

Levenberg-Marquardt

This algorithm typically requires more memory but less time. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Train using Levenberg-Marquardt. (trainlm)

Retrain

Results

	Samples	MSE	R
Training:	42	5.76484e-5	9.99498e-1
Validation:	5	4.64168e-3	9.99981e-1
Testing:	5	2.08218e-2	9.99286e-1

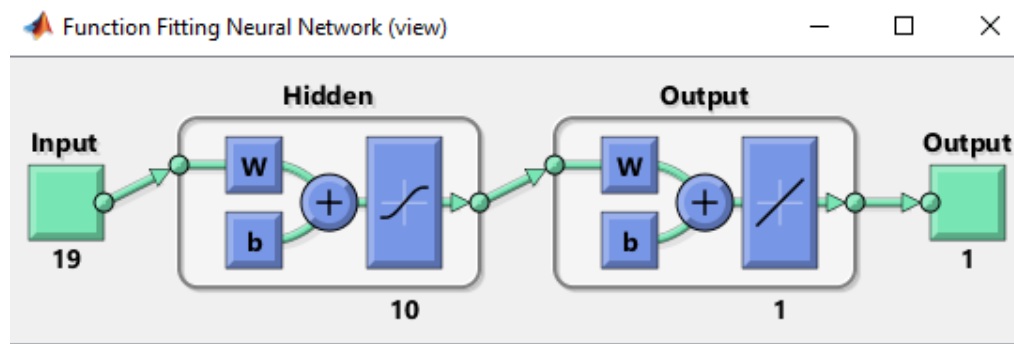
Plot Fit Plot Error Histogram Plot Regression

Notes

Training multiple times will generate different results due to different initial conditions and sampling.

Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.

Regression R Values measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship.



Función en Matlab

```
function [Y,Xf,Af] = myNeuralNetworkFunction(X,~,~)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 02-Oct-2020
13:43:25.
%
% [Y] = myNeuralNetworkFunction(X,~,~) takes these arguments:
%
% X = 1xTS cell, 1 inputs over TS timesteps
% Each X{1,ts} = 19xQ matrix, input #1 at timestep ts.
%
% and returns:
%
% Y = 1xTS cell of 1 outputs over TS timesteps.
% Each Y{1,ts} = 1xQ matrix, output #1 at timestep ts.
%
```



```
% where Q is number of samples (or series) and TS is the number of timesteps.
```

```
%#ok<*RPMTO>
```

```
% ===== NEURAL NETWORK CONSTANTS =====
```

```
% Input 1
```

```
x1_step1.xoffset = [0.004004;25.126965;9.1e-05;2.569164;0;0;0.000115;0.000181;0.000227;6.1e-05;6.9e-05;4.5e-05;0;0.000188;0;0;-3.608012;2.732656;2.72937];
```

```
x1_step1.gain = [58.0956253994074;0.00178759870523439;267.773463649752;0.00673261927038699;0.000311915159076731;0.000280583613916947;94.3307235166494;135.126004999662;47.6190476190476;781.860828772479;361.271676300578;530.363298859719;15.8165282720443;0.220065894330739;4405.28634361234;0.000311817898347365;0.370710462895034;10.6499675175991;10.6589921922882];
```

```
x1_step1.ymin = -1;
```

```
% Layer 1
```

```
b1 = [1.5787829251018566;-1.2875361156044334;0.98149391016975263;-0.48839699890741295;-0.02962056701908836;0.059678702475978;-0.34186417862318763;-0.86946859797483722;1.2208494504014762;-1.7260875721372144];
```

```
IW1_1 = [-0.5054985492743308 0.03016804674736424 0.61984228184548096 -0.038990479725908259 -0.45023587995298303 0.45718550701477556 0.25399479633570327 0.217875865009694 -0.34843758057844487 0.055666899793846057 -0.0057671834636272325 0.2107942682737845 0.31961504731647578 0.15412860621402624 -0.29024196758293513 0.068337451788598305 0.53440406585537914 0.66254025454318277 -0.16127874243365786;0.11685295445014067 -0.027378270523091643 0.42359891355704798 -0.015967122463089234 -0.30009441011370142 0.38472456612955763 -0.43082141849367422 -0.27508196760014009 -0.37787658950947067 0.09375862425475387 -0.15080436233886024 0.45774659289516106 -0.36463492742653991 -0.2828417270811181 -0.46388277000804395 0.47305105807023518 0.38238193413805377 -0.46772488869009443 -0.41665402038470928;-0.099709726620183792 0.12354744605283263 0.25749603718636416 0.033368461285478074 -
```

0.30987390503760842	0.52568630620196111	0.59280678985182256	
0.55792458447813909	-0.51792597846357857	0.12937069718284314	-
0.13765144377244259	-0.32446499140541468	-0.2003059343930188	-
0.066132117396223419	-0.057803397187783198	0.30470258919262416	
0.6785825216288871	0.32247208628393137		-
0.55069133646198631;0.18828440572431179		0.6021986213435524	
0.18265431842254753	0.078169987719333395	0.70689198497032579	
0.1776909657636524	-0.22583082570710925	-0.32803426130944657	-
0.15468804480032927	0.6084402512256688	-0.25156884749783659	-
0.32508621631300721	0.086198317683701162	0.62102783788321303	
0.49219308275242446	0.30938300768829469	0.30701143493336702	-
0.44018386266194331	0.42649648322778244;0.51303388390916604		-
0.15291959198786387	0.050334558448111924	-0.10714055656051939	-
0.18970347535513662	-0.54441049556010379	-0.11873338804505532	-
0.31131454327595887	0.44235023806837698	-0.67136056656791276	-
0.10462075764512828	-0.10723263399205787	0.79278702803192291	-
0.8046106555091227	0.51619015606649621	-0.19710361479747496	
0.23551268716832952	-0.17128729239037202		-
0.28899450991666342;0.20614692968592385		0.098766333225918365	
0.42327119734222679	0.25641729231927607	0.54762264583136089	-
0.39226743166049138	-0.01216137825052198	-0.56208383931412298	-
0.37270823771223166	-0.21463882757704883	-0.11347927560781458	
0.46483047825826601	0.20820012607471208	0.0031687733531008209	-
0.08536020894786106	-0.37842133947730949	0.65325384442884205	-
0.33580613527026715	-0.17650329372681239;-0.59252075523500192		-
0.24379505426175041	-0.13692073936611207	0.04662986033931401	-
0.15238330620963902	-0.57422377348814624	0.23451189207376105	-
0.22398112121772665	-0.65181177144983193	-0.51058443358188543	-
0.53389215732864537	-0.25463034797343287	0.18965553539870028	-
0.042094519888646778	-0.39285321965043363	-0.20606757620882593	
0.78569857019256917	0.61689702948409675	0.1905394660577388;-	
0.018722076966681891	0.58577427390653658	-0.23954793366484906	-
0.18818311757149339	-0.55410314702097796	0.50941825029792265	
0.61344830009586615	0.10168480118651002	0.32086147552163485	-
0.058408645379572958	-0.0062599968967693015	0.089986059995565942	-
0.5071912938324129	-0.019759452749685293	-0.27855025988008592	
0.43907970437261895	-0.38024881126739285	-0.35701299721013607	-
0.064723001417698448;0.70281562881507498		-0.21876809923409379	

```

0.25734631918143014      -0.1823783348522377      0.31873570192773154
0.42639649017061232      -0.19779258950910683      0.73139517794181941
0.64635147754585509      0.43524857855052063      -0.18245239454813972
0.46356150516946093      0.30755785690246235      0.5639012809400491
0.14409309540835885      0.49212279555202565      0.65141525812104817
0.69127514780249477      0.15037025088735234;-0.10652742119342296      -
0.045905682925752408      0.022378468577677273      0.5717393201618437      -
0.022633961228538148      -0.2553725299840513      0.58921978297477162      -
0.19210896199704328      0.5869022038154138      -0.65106655474133601
0.095171158725568694      0.018320816466523039      0.11481313278399095      -
0.28006184866953848      0.36062543839562627      0.56455319466990805      -
0.17932272639364102 -0.42391567249572776 -0.49666848943843156];

```

```
% Layer 2
```

```

b2 = -0.083226698844703462;
LW2_1 = [0.2161074630868102  0.12400904041810371  -0.18974956903088097
0.13082270116325878      0.30854602064679448      -0.037531214078114231      -
1.2621782915013184      -0.011888739890078576      -1.0135637561101996      -
1.2331157179788272];

```

```
% Output 1
```

```

y1_step1.ymin = -1;
y1_step1.gain = 2;
y1_step1.xoffset = 1;

```

```
% ===== SIMULATION =====
```

```
% Format Input Arguments
```

```

isCellX = iscell(X);
if ~isCellX
    X = {X};
end

```

```
% Dimensions
```

```

TS = size(X,2); % timesteps
if ~isempty(X)
    Q = size(X{1},2); % samples/series

```

```

else
    Q = 0;
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

    % Input 1
    Xp1 = mapminmax_apply(X{1,ts},x1_step1);

    % Layer 1
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);

    % Layer 2
    a2 = repmat(b2,1,Q) + LW2_1*a1;

    % Output 1
    Y{1,ts} = mapminmax_reverse(a2,y1_step1);
end

% Final Delay States
Xf = cell(1,0);
Af = cell(2,0);

% Format Output Arguments
if ~isCellX
    Y = cell2mat(Y);
end
end

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function

```

```

function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
x = bsxfun(@minus,y,settings.ymin);
x = bsxfun(@rdivide,x,settings.gain);
x = bsxfun(@plus,x,settings.xoffset);
end

```

Matlab matrix only function

```

function [y1] = myNeuralNetworkFunction(x1)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 02-Oct-2020
13:43:47.
%
% [y1] = myNeuralNetworkFunction(x1) takes these arguments:
%   x = 19xQ matrix, input #1
% and returns:
%   y = 1xQ matrix, output #1
% where Q is the number of samples.

%#ok<*RPMTO>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1

```

```

x1_step1.xoffset          =          [0.004004;25.126965;9.1e-
05;2.569164;0;0;0.000115;0.000181;0.000227;6.1e-05;6.9e-05;4.5e-
05;0;0.000188;0;0;-3.608012;2.732656;2.72937];
x1_step1.gain              =
[58.0956253994074;0.00178759870523439;267.773463649752;0.0067326192703869
9;0.000311915159076731;0.000280583613916947;94.3307235166494;135.12600499
9662;47.6190476190476;781.860828772479;361.271676300578;530.363298859719;
15.8165282720443;0.220065894330739;4405.28634361234;0.000311817898347365;
0.370710462895034;10.6499675175991;10.6589921922882];
x1_step1.ymin = -1;

```

```
% Layer 1
```

```

b1      =      [1.5787829251018566;-1.2875361156044334;0.98149391016975263;-
0.48839699890741295;-0.02962056701908836;0.059678702475978;-
0.34186417862318763;-0.86946859797483722;1.2208494504014762;-
1.7260875721372144];
IW1_1   =   [-0.5054985492743308  0.03016804674736424  0.61984228184548096  -
0.038990479725908259          -0.45023587995298303          0.45718550701477556
0.25399479633570327          0.217875865009694          -0.34843758057844487
0.055666899793846057          -0.0057671834636272325          0.2107942682737845
0.31961504731647578          0.15412860621402624          -0.29024196758293513
0.068337451788598305          0.53440406585537914          0.66254025454318277  -
0.16127874243365786;0.11685295445014067          -0.027378270523091643
0.42359891355704798          -0.015967122463089234          -0.30009441011370142
0.38472456612955763          -0.43082141849367422          -0.27508196760014009  -
0.37787658950947067          0.09375862425475387          -0.15080436233886024
0.45774659289516106          -0.36463492742653991          -0.2828417270811181  -
0.46388277000804395          0.47305105807023518          0.38238193413805377  -
0.46772488869009443          -0.41665402038470928;-0.099709726620183792  -
0.12354744605283263          0.25749603718636416          0.033368461285478074  -
0.30987390503760842          0.52568630620196111          0.59280678985182256
0.55792458447813909          -0.51792597846357857          0.12937069718284314  -
0.13765144377244259          -0.32446499140541468          -0.2003059343930188  -
0.066132117396223419          -0.057803397187783198          0.30470258919262416
0.6785825216288871          0.32247208628393137          -
0.55069133646198631;0.18828440572431179          0.6021986213435524
0.18265431842254753          0.078169987719333395          0.70689198497032579
0.1776909657636524          -0.22583082570710925          -0.32803426130944657  -

```

0.15468804480032927	0.6084402512256688	-0.25156884749783659	-
0.32508621631300721	0.086198317683701162	0.62102783788321303	
0.49219308275242446	0.30938300768829469	0.30701143493336702	-
0.44018386266194331	0.42649648322778244;	0.51303388390916604	-
0.15291959198786387	0.050334558448111924	-0.10714055656051939	-
0.18970347535513662	-0.54441049556010379	-0.11873338804505532	-
0.31131454327595887	0.44235023806837698	-0.67136056656791276	-
0.10462075764512828	-0.10723263399205787	0.79278702803192291	-
0.8046106555091227	0.51619015606649621	-0.19710361479747496	
0.23551268716832952		-0.17128729239037202	-
0.28899450991666342;	0.20614692968592385	0.098766333225918365	
0.42327119734222679	0.25641729231927607	0.54762264583136089	-
0.39226743166049138	-0.01216137825052198	-0.56208383931412298	-
0.37270823771223166	-0.21463882757704883	-0.11347927560781458	
0.46483047825826601	0.20820012607471208	0.0031687733531008209	-
0.08536020894786106	-0.37842133947730949	0.65325384442884205	-
0.33580613527026715	-0.17650329372681239;	-0.59252075523500192	-
0.24379505426175041	-0.13692073936611207	0.04662986033931401	-
0.15238330620963902	-0.57422377348814624	0.23451189207376105	-
0.22398112121772665	-0.65181177144983193	-0.51058443358188543	-
0.53389215732864537	-0.25463034797343287	0.18965553539870028	-
0.042094519888646778	-0.39285321965043363	-0.20606757620882593	
0.78569857019256917	0.61689702948409675	0.1905394660577388;	-
0.018722076966681891	0.58577427390653658	-0.23954793366484906	-
0.18818311757149339	-0.55410314702097796	0.50941825029792265	
0.61344830009586615	0.10168480118651002	0.32086147552163485	-
0.058408645379572958	-0.0062599968967693015	0.089986059995565942	-
0.5071912938324129	-0.019759452749685293	-0.27855025988008592	
0.43907970437261895	-0.38024881126739285	-0.35701299721013607	-
0.064723001417698448;	0.70281562881507498	-0.21876809923409379	
0.25734631918143014	-0.1823783348522377	0.31873570192773154	
0.42639649017061232	-0.19779258950910683	0.73139517794181941	
0.64635147754585509	0.43524857855052063	-0.18245239454813972	
0.46356150516946093	0.30755785690246235	0.5639012809400491	
0.14409309540835885	0.49212279555202565	0.65141525812104817	
0.69127514780249477	0.15037025088735234;	-0.10652742119342296	-
0.045905682925752408	0.022378468577677273	0.5717393201618437	-
0.022633961228538148	-0.2553725299840513	0.58921978297477162	-

```

0.19210896199704328      0.5869022038154138      -0.65106655474133601
0.095171158725568694    0.018320816466523039    0.11481313278399095    -
0.28006184866953848    0.36062543839562627    0.56455319466990805    -
0.17932272639364102 -0.42391567249572776 -0.49666848943843156];

```

```
% Layer 2
```

```

b2 = -0.083226698844703462;
LW2_1 = [0.2161074630868102  0.12400904041810371  -0.18974956903088097
0.13082270116325878      0.30854602064679448      -0.037531214078114231    -
1.2621782915013184      -0.011888739890078576    -1.0135637561101996    -
1.2331157179788272];

```

```
% Output 1
```

```

y1_step1.ymin = -1;
y1_step1.gain = 2;
y1_step1.xoffset = 1;

```

```
% ===== SIMULATION =====
```

```
% Dimensions
```

```
Q = size(x1,2); % samples
```

```
% Input 1
```

```
xp1 = mapminmax_apply(x1,x1_step1);
```

```
% Layer 1
```

```
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);
```

```
% Layer 2
```

```
a2 = repmat(b2,1,Q) + LW2_1*a1;
```

```
% Output 1
```

```
y1 = mapminmax_reverse(a2,y1_step1);
```

```
end
```

```
% ===== MODULE FUNCTIONS =====
```



```

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x, settings)
y = bsxfun(@minus, x, settings.xoffset);
y = bsxfun(@times, y, settings.gain);
y = bsxfun(@plus, y, settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n, ~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y, settings)
x = bsxfun(@minus, y, settings.ymin);
x = bsxfun(@rdivide, x, settings.gain);
x = bsxfun(@plus, x, settings.xoffset);
end

```

Simple script

```

% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by Neural Fitting app
% Created 02-Oct-2020 13:46:45
%
% This script assumes these variables are defined:
%
%   EMGTrainset - input data.
%   EMGTargets  - target data.

x = EMGTrainset;
t = EMGTargets;

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.

```

```
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize,trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 80/100;
net.divideParam.valRatio = 10/100;
net.divideParam.testRatio = 10/100;

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)
```

Pruebas 70% entrenamiento, 20% validación, 10% pruebas con 52 muestras

Neural Fitting (nftool) - □ ×

Validation and Test Data

Set aside some samples for validation and testing.

Select Percentages		Explanation	
Randomly divide up the 52 samples:			
Training:	70%	37 samples	Three Kinds of Samples: Training: These are presented to the network during training, and the network is adjusted according to its error. Validation: These are used to measure network generalization, and to halt training when generalization stops improving. Testing: These have no effect on training and so provide an independent measure of network performance during and after training.
Validation:	20%	10 samples	
Testing:	10%	5 samples	

Network Architecture

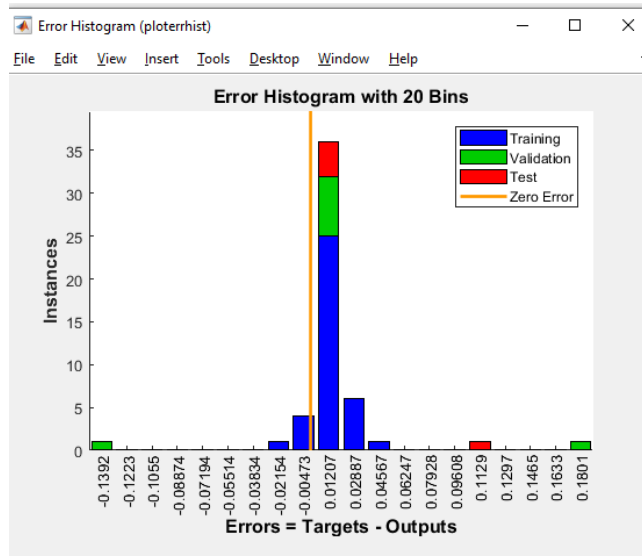
Set the number of neurons in the fitting network's hidden layer.

Hidden Layer	Recommendation
Define a fitting neural network. (fitnet) Number of Hidden Neurons: <input type="text" value="10"/> <input type="button" value="Restore Defaults"/>	Return to this panel and change the number of neurons if the network does not perform well after training.

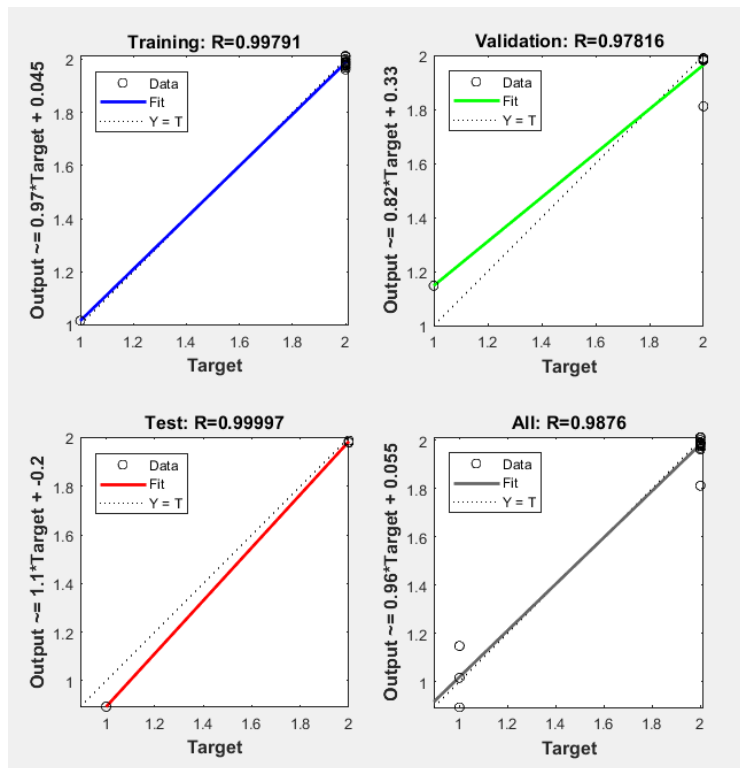
Neural Network

The diagram illustrates a neural network with the following structure:

- Input Layer:** 19 neurons.
- Hidden Layer:** 10 neurons. Each neuron receives input from the input layer, is multiplied by a weight matrix W , a bias matrix b is added, and the result passes through an activation function.
- Output Layer:** 1 neuron. It receives input from the hidden layer, is multiplied by a weight matrix W , a bias matrix b is added, and the result passes through an activation function.



Histograma de error



Gráfica de regresión

Función en matlab

```

function [Y,Xf,Af] = myNeuralNetworkFunction(X,~,~)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 02-Oct-2020
13:56:40.
%
% [Y] = myNeuralNetworkFunction(X,~,~) takes these arguments:
%
% X = 1xTS cell, 1 inputs over TS timesteps
% Each X{1,ts} = 19xQ matrix, input #1 at timestep ts.
%
% and returns:
% Y = 1xTS cell of 1 outputs over TS timesteps.
% Each Y{1,ts} = 1xQ matrix, output #1 at timestep ts.
%
% where Q is number of samples (or series) and TS is the number of timesteps.

%#ok<*RPMTO>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset = [0.004004;25.126965;9.1e-
05;2.569164;0;0;0.000115;0.000181;0.000227;6.1e-05;6.9e-05;4.5e-
05;0;0.000188;0;0;-3.608012;2.732656;2.72937];
x1_step1.gain =
[58.0956253994074;0.00178759870523439;267.773463649752;0.0067326192703869
9;0.000311915159076731;0.000280583613916947;94.3307235166494;135.12600499
9662;47.6190476190476;781.860828772479;361.271676300578;530.363298859719;
15.8165282720443;0.220065894330739;4405.28634361234;0.000311817898347365;
0.370710462895034;10.6499675175991;10.6589921922882];
x1_step1.ymin = -1;

% Layer 1

```

```

b1      =      [1.5761023600475226;1.1571753327852845;-0.81698452181099734;-
0.74959766871240519;0.058309151405422338;0.35407608129596052;-
0.55510668191709944;-1.0657261819139781;-
1.1569957661920225;1.5796584523833976];
IW1_1   =   [-0.29794349850225332   -0.10268130369261193   0.80195599557248876
0.29809438580971181       -0.45381578509568588       0.23595188549576079       -
0.47218554984599254       -0.1956898104252443       0.28400570469732406
0.55669958227138983       -0.33044104694653076       0.21051358295314429       -
0.10730580148450135       -0.018459109246913588       -0.034197470335609999       -
0.41878893415365248       0.11727150529190561       -0.26048937936711042       -
0.18054465470031025;-0.86709062482780042       0.19676858910714431
0.18482357194745674       -0.039647994201775318       -0.11487539187900532       -
0.068490904743528563       -0.18789880587546909       0.28868443404413036
0.27687179316825983       -0.96730903120857348       -0.64098063915423786       -
0.33007043161051297       0.30377265073991377       0.38826249368006388
0.032322795678857874       0.11914567559316039       -0.33133536581857814       -
0.011830784403428149       -1.0060182072058381;-0.11203909816914254       -
0.29825735837584716       0.27793819991114327       -0.33671459968973999       -
0.53536123924209311       0.064723681357904642       -0.6546738435424474
0.32045931818082135       0.023957031737953216       -0.62257006559755845       -
0.46945587246946296       -0.43019052830866966       -0.52079920212803998
0.40969841311552396       0.37121261310649811       -0.46447141504514222
0.25437062326192894       -0.27588869444595837
0.33801741711869487;0.47629941758953653       -0.39519956199831208       -
0.20680883365356978       -0.19555197621921991       0.40185754124299905
0.60153720226625673       -0.12522807405152203       0.58902847260175661
0.13930036763650008       0.34094479468237038       0.37469551071978169
0.7048476087793063       0.86302564540467941       0.48279765325510043       -
0.15892264162822153       0.0014569447038535405       0.402788666608555       -
0.46712648446970584       0.053256311473997862;-0.48424534381526424
0.075668253897155771       -0.22051584213245853       -0.35910249627549368
0.38555518460425875       -0.54080347856853828       -0.025545163052552236       -
0.3156880977567148       -0.32705740153473684       0.75523808232650202
0.31763896611369952       0.16080302128859136       0.7097625156563182       -
0.10903223577550683       -0.17061461472352846       -0.26480092508333586
0.025393641817389609       0.51909561525458647
0.091738780476786366;0.21696316269273219       0.38172041733939566       -
0.47218623901477402       0.14494015121743781       0.68043187685564011       -

```

```

0.14520392269410454    0.35938605181253908    -0.20927472262127816    -
0.57730503495305663    0.48578780130449184    0.28844031430419131    -
0.19237392758601379    -0.67208733867557646    -0.43145578658040834    -
0.51886571877920662    0.45096183459813471    0.28241474772281683
0.2211398448978594    0.48480994727511634;-0.31376653103855207
0.095154563708370124    0.62188971803214654    -0.42490240013080527    -
0.56039791579936693    0.18715533590469055    -0.35846479994634728    -
0.33237732007842075    -0.17887047700499312    0.23546341614659591
0.26736707452792291    0.094489706110846228    0.54787345771143814    -
0.28020575565176614    0.58331290826788529    -0.65790026770685217    -
0.17823787044045336    0.27975077078621791    -0.01565682529244785;-
0.34691275858507414    0.18219957621162475    0.57540851823583394    -
0.15179642271266211    0.55486030931752894    -0.399649029554556
0.4140293506029023    -0.51093610979467408    -0.20192751314503174
0.070000126833305254    0.51001945796141934    0.23152841804427793
0.2247593900846282    0.30076581762441201    0.11472551870694463
0.21869541183147867    0.22865987255110681    -0.7005973400077572    -
0.68442887401129993;-0.31752299281771057    -0.57933504629083132
0.20350257505438013    -0.090584294550025327    0.52092449517847383
0.39719071877778817    -0.039175534876430912    -0.50697548542782533    -
0.41261973504469973    -0.60216311848619919    0.24920832368028401
0.39821485735614248    -0.022869286282741121    0.26188888612838002
0.43425666404943331    -0.33756188933503189    -0.012227313564029826    -
0.052080447979822052    -0.44992254094182343;0.42605051237562636
0.031098839856413464    0.061055380315287164    0.14320400954089474
0.31438607552166625    0.50755854559258895    0.1403634370468331
0.45623246858635969    -0.097478701387326006    -0.53463836811171328    -
0.29606454205144961    0.48536820792320073    0.54582956848148723
0.18072414294627703    -0.67856170577359143    0.17050363590526915
0.10268256814521487 0.33877317747507885 -0.33196639167248515];

```

```
% Layer 2
```

```
b2 = 0.57885637062156814;
```

```

LW2_1 = [0.74049677116310308    0.98962654400408967    0.32779042193820568
1.5210362899537466    -0.44227465544514905    0.47906395538515567
0.53836667291661311    -0.30053466441796334    0.56100570879569234
0.022523806880672147];

```

```
% Output 1
y1_step1.ymin = -1;
y1_step1.gain = 2;
y1_step1.xoffset = 1;

% ===== SIMULATION =====

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX
    X = {X};
end

% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
    Q = size(X{1},2); % samples/series
else
    Q = 0;
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

    % Input 1
    Xp1 = mapminmax_apply(X{1,ts},x1_step1);

    % Layer 1
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);

    % Layer 2
    a2 = repmat(b2,1,Q) + LW2_1*a1;
```



```

    % Output 1
    Y{1,ts} = mapminmax_reverse(a2,y1_step1);
end

% Final Delay States
Xf = cell(1,0);
Af = cell(2,0);

% Format Output Arguments
if ~isCellX
    Y = cell2mat(Y);
end
end

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x, settings)
y = bsxfun(@minus, x, settings.xoffset);
y = bsxfun(@times, y, settings.gain);
y = bsxfun(@plus, y, settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n, ~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y, settings)
x = bsxfun(@minus, y, settings.ymin);
x = bsxfun(@rdivide, x, settings.gain);
x = bsxfun(@plus, x, settings.xoffset);
end

```

Matlab matrix only function

```

function [y1] = myNeuralNetworkFunction(x1)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 02-Oct-2020
13:56:58.
%
% [y1] = myNeuralNetworkFunction(x1) takes these arguments:
%   x = 19xQ matrix, input #1
% and returns:
%   y = 1xQ matrix, output #1
% where Q is the number of samples.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset = [0.004004;25.126965;9.1e-
05;2.569164;0;0;0.000115;0.000181;0.000227;6.1e-05;6.9e-05;4.5e-
05;0;0.000188;0;0;-3.608012;2.732656;2.72937];
x1_step1.gain =
[58.0956253994074;0.00178759870523439;267.773463649752;0.0067326192703869
9;0.000311915159076731;0.000280583613916947;94.3307235166494;135.12600499
9662;47.6190476190476;781.860828772479;361.271676300578;530.363298859719;
15.8165282720443;0.220065894330739;4405.28634361234;0.000311817898347365;
0.370710462895034;10.6499675175991;10.6589921922882];
x1_step1.ymin = -1;

% Layer 1
b1 = [1.5761023600475226;1.1571753327852845;-0.81698452181099734;-
0.74959766871240519;0.058309151405422338;0.35407608129596052;-
0.55510668191709944;-1.0657261819139781;-
1.1569957661920225;1.5796584523833976];
IW1_1 = [-0.29794349850225332 -0.10268130369261193 0.80195599557248876
0.29809438580971181 -0.45381578509568588 0.23595188549576079 -
0.47218554984599254 -0.1956898104252443 0.28400570469732406
0.55669958227138983 -0.33044104694653076 0.21051358295314429 -

```

0.10730580148450135	-0.018459109246913588	-0.034197470335609999	-
0.41878893415365248	0.11727150529190561	-0.26048937936711042	-
0.18054465470031025;	-0.86709062482780042	0.19676858910714431	
0.18482357194745674	-0.039647994201775318	-0.11487539187900532	-
0.068490904743528563	-0.18789880587546909	0.28868443404413036	
0.27687179316825983	-0.96730903120857348	-0.64098063915423786	-
0.33007043161051297	0.30377265073991377	0.38826249368006388	
0.032322795678857874	0.11914567559316039	-0.33133536581857814	-
0.011830784403428149	-1.0060182072058381;	-0.11203909816914254	-
0.29825735837584716	0.27793819991114327	-0.33671459968973999	-
0.53536123924209311	0.064723681357904642	-0.6546738435424474	
0.32045931818082135	0.023957031737953216	-0.62257006559755845	-
0.46945587246946296	-0.43019052830866966	-0.52079920212803998	
0.40969841311552396	0.37121261310649811	-0.46447141504514222	
0.25437062326192894		-0.27588869444595837	
0.33801741711869487;	0.47629941758953653	-0.39519956199831208	-
0.20680883365356978	-0.19555197621921991	0.40185754124299905	
0.60153720226625673	-0.12522807405152203	0.58902847260175661	
0.13930036763650008	0.34094479468237038	0.37469551071978169	
0.7048476087793063	0.86302564540467941	0.48279765325510043	-
0.15892264162822153	0.0014569447038535405	0.402788666608555	-
0.46712648446970584	0.053256311473997862;	-0.48424534381526424	
0.075668253897155771	-0.22051584213245853	-0.35910249627549368	
0.38555518460425875	-0.54080347856853828	-0.025545163052552236	-
0.3156880977567148	-0.32705740153473684	0.75523808232650202	
0.31763896611369952	0.16080302128859136	0.7097625156563182	-
0.10903223577550683	-0.17061461472352846	-0.26480092508333586	
0.025393641817389609		0.51909561525458647	
0.091738780476786366;	0.21696316269273219	0.38172041733939566	-
0.47218623901477402	0.14494015121743781	0.68043187685564011	-
0.14520392269410454	0.35938605181253908	-0.20927472262127816	-
0.57730503495305663	0.48578780130449184	0.28844031430419131	-
0.19237392758601379	-0.67208733867557646	-0.43145578658040834	-
0.51886571877920662	0.45096183459813471	0.28241474772281683	
0.2211398448978594	0.48480994727511634;	-0.31376653103855207	
0.095154563708370124	0.62188971803214654	-0.42490240013080527	-
0.56039791579936693	0.18715533590469055	-0.35846479994634728	-
0.33237732007842075	-0.17887047700499312	0.23546341614659591	

```

0.26736707452792291    0.094489706110846228    0.54787345771143814    -
0.28020575565176614    0.58331290826788529    -0.65790026770685217    -
0.17823787044045336    0.27975077078621791    -0.01565682529244785;-
0.34691275858507414    0.18219957621162475    0.57540851823583394    -
0.15179642271266211    0.55486030931752894    -0.399649029554556
0.4140293506029023    -0.51093610979467408    -0.20192751314503174
0.070000126833305254    0.51001945796141934    0.23152841804427793
0.2247593900846282    0.30076581762441201    0.11472551870694463
0.21869541183147867    0.22865987255110681    -0.7005973400077572    -
0.68442887401129993;-0.31752299281771057    -0.57933504629083132
0.20350257505438013    -0.090584294550025327    0.52092449517847383
0.39719071877778817    -0.039175534876430912    -0.50697548542782533    -
0.41261973504469973    -0.60216311848619919    0.24920832368028401
0.39821485735614248    -0.022869286282741121    0.26188888612838002
0.43425666404943331    -0.33756188933503189    -0.012227313564029826    -
0.052080447979822052    -0.44992254094182343;0.42605051237562636
0.031098839856413464    0.061055380315287164    0.14320400954089474
0.31438607552166625    0.50755854559258895    0.1403634370468331
0.45623246858635969    -0.097478701387326006    -0.53463836811171328    -
0.29606454205144961    0.48536820792320073    0.54582956848148723
0.18072414294627703    -0.67856170577359143    0.17050363590526915
0.10268256814521487 0.33877317747507885 -0.33196639167248515];

```

```
% Layer 2
```

```

b2 = 0.57885637062156814;
LW2_1 = [0.74049677116310308    0.98962654400408967    0.32779042193820568
1.5210362899537466    -0.44227465544514905    0.47906395538515567
0.53836667291661311    -0.30053466441796334    0.56100570879569234
0.022523806880672147];

```

```
% Output 1
```

```

y1_step1.ymin = -1;
y1_step1.gain = 2;
y1_step1.xoffset = 1;

```

```
% ===== SIMULATION =====
```

```

% Dimensions
Q = size(x1,2); % samples

% Input 1
xp1 = mapminmax_apply(x1,x1_step1);

% Layer 1
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);

% Layer 2
a2 = repmat(b2,1,Q) + LW2_1*a1;

% Output 1
y1 = mapminmax_reverse(a2,y1_step1);
end

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
x = bsxfun(@minus,y,settings.ymin);
x = bsxfun(@rdivide,x,settings.gain);
x = bsxfun(@plus,x,settings.xoffset);
end

```

Simple script

```
% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by Neural Fitting app
% Created 02-Oct-2020 14:00:39
%
% This script assumes these variables are defined:
%
%   EMGTrainset - input data.
%   EMGTargets  - target data.

x = EMGTrainset;
t = EMGTargets;

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize,trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 10/100;
net.divideParam.testRatio = 20/100;

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
```

```

performance = perform(net,t,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)

```

ADVANCED SCRIPT

```

% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by Neural Fitting app
% Created 02-Oct-2020 14:01:14
%
% This script assumes these variables are defined:
%
%   EMGTrainset - input data.
%   EMGTargets  - target data.

x = EMGTrainset;
t = EMGTargets;

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 10;

```

```

net = fitnet(hiddenLayerSize,trainFcn);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand'; % Divide data randomly
net.divideMode = 'sample'; % Divide up every sample
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 10/100;
net.divideParam.testRatio = 20/100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse'; % Mean Squared Error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotregression','plotfit'};

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)

```



```
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)

% Deployment
% Change the (false) values to (true) to enable the following code blocks.
% See the help for each generation function for more information.
if (false)
    % Generate MATLAB function for neural network for application
    % deployment in MATLAB scripts or with MATLAB Compiler and Builder
    % tools, or simply to examine the calculations your trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a matrix-only MATLAB function for neural network code
    % generation with MATLAB Coder tools.
    genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end
```

Pruebas 65% entrenamiento, 20% validación, 15% pruebas con 74 muestras

Neural Fitting (nftool) [minimize] [maximize] [close]

Validation and Test Data

Set aside some samples for validation and testing.

Select Percentages

Randomly divide up the 74 samples:

Training:	65%	48 samples
Validation:	20%	15 samples
Testing:	15%	11 samples

Explanation

Three Kinds of Samples:

- Training:** These are presented to the network during training, and the network is adjusted according to its error.
- Validation:** These are used to measure network generalization, and to halt training when generalization stops improving.
- Testing:** These have no effect on training and so provide an independent measure of network performance during and after training.

Network Architecture

Set the number of neurons in the fitting network's hidden layer.

Hidden Layer

Define a fitting neural network. (fitnet)

Number of Hidden Neurons:

[Restore Defaults](#)

Recommendation

Return to this panel and change the number of neurons if the network does not perform well after training.

Neural Network

The diagram illustrates a neural network architecture with the following components:

- Input Layer:** 19 neurons.
- Hidden Layer:** 10 neurons. It includes weight (W) and bias (b) inputs, a summation node (+), and a sigmoid-shaped activation function.
- Output Layer:** 1 neuron. It includes weight (W) and bias (b) inputs, a summation node (+), and a linear-shaped activation function.

Train Network

Train the network to fit the inputs and targets.

Train Network

Choose a training algorithm:

Levenberg-Marquardt ▾

This algorithm typically requires more memory but less time. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Train using Levenberg-Marquardt. (trainlm)

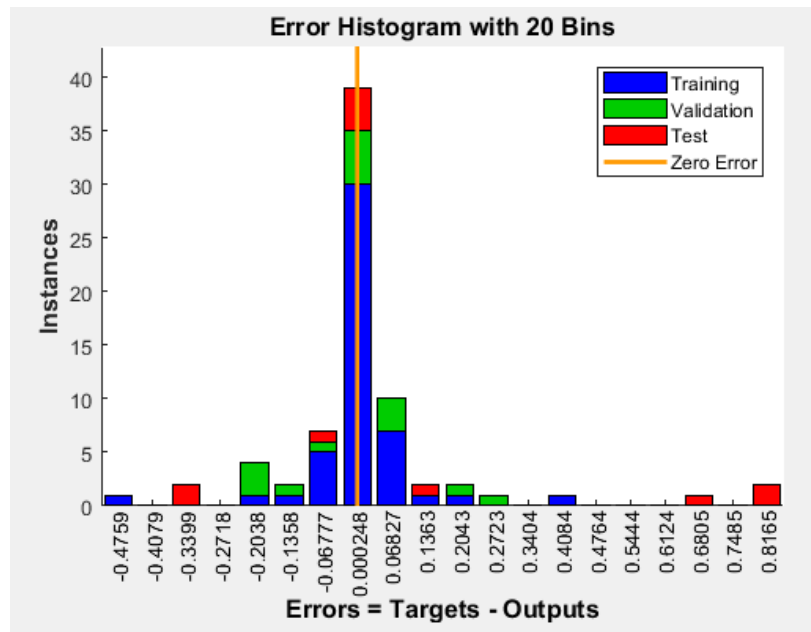
Retrain

Results

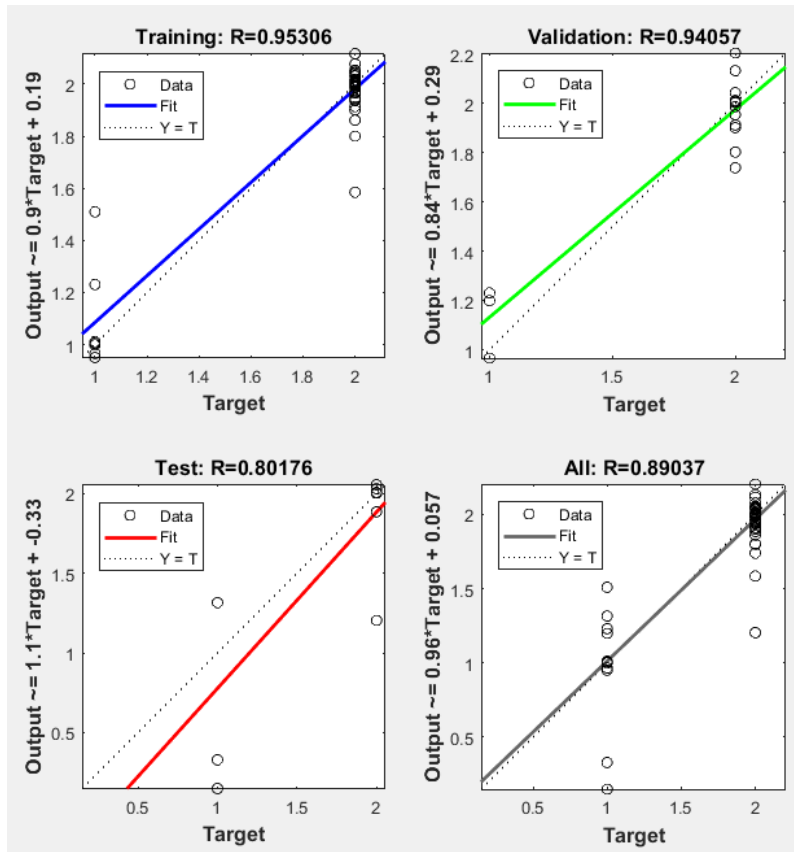
	Samples	MSE	R
Training:	48	1.27586e-2	9.53058e-1
Validation:	15	1.88086e-2	9.40568e-1
Testing:	11	1.83946e-1	8.01760e-1

Notes

- Training multiple times will generate different results due to different initial conditions and sampling.
- Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.
- Regression R Values measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship.



Histograma de error



Gráfica de regresión

Matlab function

```
function [Y,Xf,Af] = myNeuralNetworkFunction(X,~,~)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 02-Oct-2020
16:25:28.
%
% [Y] = myNeuralNetworkFunction(X,~,~) takes these arguments:
%
% X = 1xTS cell, 1 inputs over TS timesteps
% Each X{1,ts} = 19xQ matrix, input #1 at timestep ts.
%
% and returns:
% Y = 1xTS cell of 1 outputs over TS timesteps.
% Each Y{1,ts} = 1xQ matrix, output #1 at timestep ts.
%
```

```
% where Q is number of samples (or series) and TS is the number of timesteps.
```

```
%#ok<*RPMTO>
```

```
% ===== NEURAL NETWORK CONSTANTS =====
```

```
% Input 1
```

```
x1_step1.xoffset = [0.003856;25.126965;8.3e-05;2.517872;0;0;0.000104;0.000165;0.000206;5.5e-05;6.2e-05;4.1e-05;0;0.000165;0;0;-3.671084;2.732656;2.72937];
```

```
x1_step1.gain = [22.0242486978163;0.000752660216947654;63.9611116441204;0.00196899405567488;0.000119410114036659;0.000107625248883388;32.3086118604914;32.2461022523902;16.285451391999;175.947919415853;85.0448611642641;128.09017548354;4.52190978337791;0.0307555157018749;520.156046814044;0.000119338862700638;0.310411560719838;10.5815626851774;10.2376150451991];
```

```
x1_step1.ymin = -1;
```

```
% Layer 1
```

```
b1 = [-1.7101538678313501;0.8833013997558905;-1.0654623054709318;0.2848610089631074;0.19067682679521039;0.15819402116558892;-0.90861850906276231;1.5472780755428526;-1.4247361759681836;-1.7121742545393941];
```

```
IW1_1 = [0.013794847753043692 0.479475377971192 -0.043388814419406486
0.12763944812648073 0.10651885061945647 -0.36159675225500626
0.27243368295637438 -0.60721640540508548 0.2609391921591882
0.17980402877035745 -0.39904814324892152 -0.12154969384119406
0.40169366352356312 0.36759514437076629 0.44485626070559492 -
0.31793958952652945 -0.65741888970428064 0.26992554354791354 -
0.45697227099770171;-0.46272958251833579 -0.56411371288701229
0.71437373566211804 0.41717256030014582 0.26328425538006089
0.28466772381433703 0.19295839663840106 0.17194337845859814 -
0.26995473970501238 0.28798793727580707 0.25259982863883534
0.47367108733994961 0.21804711893180229 0.96490527669065473
0.55236587810552462 0.0045280967564247888 0.17713747010932193 -
0.25636717851739305 -0.27994693979400581;-1.5032811563815078
0.93767245472231964 -1.0222116650464053 -0.023008105384592123 -
```

0.1019198513065747	-0.014261140900194449	-0.44345580238195992	-
0.21431263321886221	0.63339787944467496	-1.472624706005824	
0.04208918813859882	-0.58261447335477923	-0.74458471866055354	
0.28735504602599798	0.3629379707259584	-0.50303028063444544	-
1.7816817934258904	-1.5432681660972416		-
0.65570705457603851;	0.72625200155756842	0.19393584940709138	
0.092325357168985531	-0.17635622001477033	0.89458713730324124	
0.6003469438748692	0.041785935264687907	-0.38882146480254659	-
0.43495196037259459	0.21187709623729062	-0.77491728098228729	
0.09778910626297907	-0.92380302360035416	-0.21692827148184129	-
0.85140414970632439	0.83164579119472781	1.1078528226188595	
1.9736018088101401	1.8509094686611083;	-0.13237428837277412	
1.3850169542287567	-0.096912825553089929	0.33231013148740507	-
0.26082964765537153	0.014337324491241333	-0.1411003825722876	-
0.03478151430864708	0.14839248828468957	-0.73190689670009668	
0.020086228840938771	0.3671699909134144	-0.16083343116825369	-
0.037843292322786715	0.11124664566209209	-0.37524748389612894	-
1.4968152432955308	4.0574761365336736	2.7480644037621453;	-
0.94585182947781343	-0.54174842913297372	0.19353449046102653	-
0.12286079558298064	0.23866380352608652	-0.033993637963172681	-
0.6258732267139433	-0.092389932459068458	0.22067373759860362	-
0.20110266080630515	0.1859847843695685	-0.32774010457001163	
0.40173789540813387	-0.27334846643303057	-0.31275924275235645	-
0.37098031052556441	3.2180299720021681	-0.14142127002859883	
7.7177885077412354;	-1.7729321521701558	-1.7760295972352231	
0.63436226687276998	-0.041606086020237763	0.6315807296857785	
0.66798387990013031	-0.33256029535989573	0.14625598346971258	
0.3267300346010078	0.28578986897149283	-0.077675662618256924	
0.94600638525951564	1.319604287142063	0.44326115771670671	
1.188082983316334	-0.26103767926959992	-4.3766247632283397	-
0.2590074527896094	1.0066595125579514;	-1.8384506895440778	
2.362263981651433	0.088718198920537561	-0.080266627822589962	-
0.91474739692494023	-0.092846139186339896	-0.74023826612080479	
0.10064055994380883	-0.57166207330507168	-0.79550176050703802	-
0.40915121337219046	0.40802833869001942	-0.31281496561405747	-
0.31233773231851308	-0.59166591180285366	-0.73483679758012133	
6.9773959576213445	7.5250192818127655	-6.9482024342838056;	-
1.8158872171193137	2.5010238240539255	-0.21446999060652594	-

```

0.034244532124751911    -0.59676033285612751    -0.81893553755034387    -
0.028194449381171157    -0.56148888756711979    -1.1225467636533544    -
0.90901588501806763        -0.44709453057212367        0.028530679042960572
0.00069523474169054696    -0.65591023309087559    0.17362013736864582    -
0.97205511100374686    0.78661901343277685    -3.7478317639175454    -
2.9326976637918936;-0.21892378211132754        -0.54508198669455243
0.46605821317630386        0.07288114978714294        0.17324408051013765
0.018328635875664058        0.018041985121370757        -0.25527212532018106
0.0084691709052113866        0.38358065842654743        0.25998078755172122
0.45201952053099143        -0.026325753459633117        0.40426246065071703
0.29231617194802534        0.63316788528686307        0.0068558687579522949
0.09640801873170271 0.24551228203412481];

```

```
% Layer 2
```

```

b2 = 0.8621508231042796;
LW2_1 = [-0.84853609558509635  0.14996593555651461  -3.1284331625649977  -
2.1692085340093503            -4.2656308945117836            5.1435038721949553
3.9312687803888191            2.8495018238939784            1.7569178794361546            -
0.62071065252317237];

```

```
% Output 1
```

```

y1_step1.ymin = -1;
y1_step1.gain = 2;
y1_step1.xoffset = 1;

```

```
% ===== SIMULATION =====
```

```
% Format Input Arguments
```

```

isCellX = iscell(X);
if ~isCellX
    X = {X};
end

```

```
% Dimensions
```

```

TS = size(X,2); % timesteps
if ~isempty(X)
    Q = size(X{1},2); % samples/series

```

```

else
    Q = 0;
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

    % Input 1
    Xp1 = mapminmax_apply(X{1,ts},x1_step1);

    % Layer 1
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);

    % Layer 2
    a2 = repmat(b2,1,Q) + LW2_1*a1;

    % Output 1
    Y{1,ts} = mapminmax_reverse(a2,y1_step1);
end

% Final Delay States
Xf = cell(1,0);
Af = cell(2,0);

% Format Output Arguments
if ~isCellX
    Y = cell2mat(Y);
end
end

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function

```



```

function y = mapminmax_apply(x, settings)
y = bsxfun(@minus, x, settings.xoffset);
y = bsxfun(@times, y, settings.gain);
y = bsxfun(@plus, y, settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n, ~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y, settings)
x = bsxfun(@minus, y, settings.ymin);
x = bsxfun(@rdivide, x, settings.gain);
x = bsxfun(@plus, x, settings.xoffset);
end

```

Matlab matrix only function

```

function [y1] = myNeuralNetworkFunction(x1)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 02-Oct-2020
16:25:43.
%
% [y1] = myNeuralNetworkFunction(x1) takes these arguments:
%   x = 19xQ matrix, input #1
% and returns:
%   y = 1xQ matrix, output #1
% where Q is the number of samples.

%#ok<*RPMTO>

```

```

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset          =          [0.003856;25.126965;8.3e-
05;2.517872;0;0;0.000104;0.000165;0.000206;5.5e-05;6.2e-05;4.1e-
05;0;0.000165;0;0;-3.671084;2.732656;2.72937];
x1_step1.gain              =
[22.0242486978163;0.000752660216947654;63.9611116441204;0.001968994055674
88;0.000119410114036659;0.000107625248883388;32.3086118604914;32.24610225
23902;16.285451391999;175.947919415853;85.0448611642641;128.09017548354;4
.52190978337791;0.0307555157018749;520.156046814044;0.000119338862700638;
0.310411560719838;10.5815626851774;10.2376150451991];
x1_step1.ymin = -1;

% Layer 1
b1          =          [-1.7101538678313501;0.8833013997558905;-
1.0654623054709318;0.2848610089631074;0.19067682679521039;0.1581940211655
8892;-0.90861850906276231;1.5472780755428526;-1.4247361759681836;-
1.7121742545393941];
IW1_1 = [0.013794847753043692  0.479475377971192  -0.043388814419406486
0.12763944812648073          0.10651885061945647          -0.36159675225500626
0.27243368295637438          -0.60721640540508548          0.2609391921591882
0.17980402877035745          -0.39904814324892152          -0.12154969384119406
0.40169366352356312          0.36759514437076629          0.44485626070559492  -
0.31793958952652945          -0.65741888970428064          0.26992554354791354  -
0.45697227099770171;-0.46272958251833579          -0.56411371288701229
0.71437373566211804          0.41717256030014582          0.26328425538006089
0.28466772381433703          0.19295839663840106          0.17194337845859814  -
0.26995473970501238          0.28798793727580707          0.25259982863883534
0.47367108733994961          0.21804711893180229          0.96490527669065473
0.55236587810552462          0.0045280967564247888          0.17713747010932193  -
0.25636717851739305          -0.27994693979400581;-1.5032811563815078
0.93767245472231964          -1.0222116650464053          -0.023008105384592123  -
0.1019198513065747          -0.014261140900194449          -0.44345580238195992  -
0.21431263321886221          0.63339787944467496          -1.472624706005824
0.04208918813859882          -0.58261447335477923          -0.74458471866055354
0.28735504602599798          0.3629379707259584          -0.50303028063444544  -
1.7816817934258904          -1.5432681660972416          -

```

0.65570705457603851;	0.72625200155756842	0.19393584940709138
0.092325357168985531	-0.17635622001477033	0.89458713730324124
0.6003469438748692	0.041785935264687907	-0.38882146480254659 -
0.43495196037259459	0.21187709623729062	-0.77491728098228729
0.09778910626297907	-0.92380302360035416	-0.21692827148184129 -
0.85140414970632439	0.83164579119472781	1.1078528226188595
1.9736018088101401	1.8509094686611083;	-0.13237428837277412
1.3850169542287567	-0.096912825553089929	0.33231013148740507 -
0.26082964765537153	0.014337324491241333	-0.1411003825722876 -
0.03478151430864708	0.14839248828468957	-0.73190689670009668
0.020086228840938771	0.3671699909134144	-0.16083343116825369 -
0.037843292322786715	0.11124664566209209	-0.37524748389612894 -
1.4968152432955308	4.0574761365336736	2.7480644037621453;-
0.94585182947781343	-0.54174842913297372	0.19353449046102653 -
0.12286079558298064	0.23866380352608652	-0.033993637963172681 -
0.6258732267139433	-0.092389932459068458	0.22067373759860362 -
0.20110266080630515	0.1859847843695685	-0.32774010457001163
0.40173789540813387	-0.27334846643303057	-0.31275924275235645 -
0.37098031052556441	3.2180299720021681	-0.14142127002859883
7.7177885077412354;-	1.7729321521701558	-1.7760295972352231
0.63436226687276998	-0.041606086020237763	0.6315807296857785
0.66798387990013031	-0.33256029535989573	0.14625598346971258
0.3267300346010078	0.28578986897149283	-0.077675662618256924
0.94600638525951564	1.319604287142063	0.44326115771670671
1.188082983316334	-0.26103767926959992	-4.3766247632283397 -
0.2590074527896094	1.0066595125579514;-	-1.8384506895440778
2.362263981651433	0.088718198920537561	-0.080266627822589962 -
0.91474739692494023	-0.092846139186339896	-0.74023826612080479
0.10064055994380883	-0.57166207330507168	-0.79550176050703802 -
0.40915121337219046	0.40802833869001942	-0.31281496561405747 -
0.31233773231851308	-0.59166591180285366	-0.73483679758012133
6.9773959576213445	7.5250192818127655	-6.9482024342838056;-
1.8158872171193137	2.5010238240539255	-0.21446999060652594 -
0.034244532124751911	-0.59676033285612751	-0.81893553755034387 -
0.028194449381171157	-0.56148888756711979	-1.1225467636533544 -
0.90901588501806763	-0.44709453057212367	0.028530679042960572
0.00069523474169054696	-0.65591023309087559	0.17362013736864582 -
0.97205511100374686	0.78661901343277685	-3.7478317639175454 -

```

2.9326976637918936;-0.21892378211132754          -0.54508198669455243
0.46605821317630386          0.07288114978714294          0.17324408051013765
0.018328635875664058          0.018041985121370757          -0.25527212532018106
0.0084691709052113866          0.38358065842654743          0.25998078755172122
0.45201952053099143          -0.026325753459633117          0.40426246065071703
0.29231617194802534          0.63316788528686307          0.0068558687579522949
0.09640801873170271 0.24551228203412481];

```

```
% Layer 2
```

```
b2 = 0.8621508231042796;
```

```

LW2_1 = [-0.84853609558509635  0.14996593555651461  -3.1284331625649977  -
2.1692085340093503          -4.2656308945117836          5.1435038721949553
3.9312687803888191          2.8495018238939784          1.7569178794361546          -
0.62071065252317237];

```

```
% Output 1
```

```
y1_step1.ymin = -1;
```

```
y1_step1.gain = 2;
```

```
y1_step1.xoffset = 1;
```

```
% ===== SIMULATION =====
```

```
% Dimensions
```

```
Q = size(x1,2); % samples
```

```
% Input 1
```

```
xp1 = mapminmax_apply(x1,x1_step1);
```

```
% Layer 1
```

```
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);
```

```
% Layer 2
```

```
a2 = repmat(b2,1,Q) + LW2_1*a1;
```

```
% Output 1
```

```
y1 = mapminmax_reverse(a2,y1_step1);
```

```
end
```

```

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x, settings)
y = bsxfun(@minus, x, settings.xoffset);
y = bsxfun(@times, y, settings.gain);
y = bsxfun(@plus, y, settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n, ~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y, settings)
x = bsxfun(@minus, y, settings.ymin);
x = bsxfun(@rdivide, x, settings.gain);
x = bsxfun(@plus, x, settings.xoffset);
end

```

Script

```

% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by Neural Fitting app
% Created 02-Oct-2020 16:27:26
%
% This script assumes these variables are defined:
%
%   EMGFullldata - input data.
%   EMGTargetFullldata - target data.

x = EMGFullldata;
t = EMGTargetFullldata;

% Choose a Training Function

```

```
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize,trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 65/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 20/100;

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)
```

Advanced script

```

% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by Neural Fitting app
% Created 02-Oct-2020 16:28:01
%
% This script assumes these variables are defined:
%
%   EMGFullldata - input data.
%   EMGTargetFullldata - target data.

x = EMGFullldata;
t = EMGTargetFullldata;

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize,trainFcn);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand'; % Divide data randomly
net.divideMode = 'sample'; % Divide up every sample
net.divideParam.trainRatio = 65/100;
net.divideParam.valRatio = 15/100;

```

```
net.divideParam.testRatio = 20/100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse'; % Mean Squared Error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
               'plotregression', 'plotfit'};

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)
```



```
% Deployment
% Change the (false) values to (true) to enable the following code blocks.
% See the help for each generation function for more information.
if (false)
    % Generate MATLAB function for neural network for application
    % deployment in MATLAB scripts or with MATLAB Compiler and Builder
    % tools, or simply to examine the calculations your trained neural
    % network performs.
    genFunction(net, 'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a matrix-only MATLAB function for neural network code
    % generation with MATLAB Coder tools.
    genFunction(net, 'myNeuralNetworkFunction', 'MatrixOnly', 'yes');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end
```

Pruebas 65% entrenamiento, 20% validación, 15% pruebas con 74 muestras y 16 neuronas

Neural Fitting (nftool)
— □ ×

Train Network

Train the network to fit the inputs and targets.

Train Network

Choose a training algorithm:

Levenberg-Marquardt ▾

This algorithm typically requires more memory but less time. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Train using Levenberg-Marquardt. (trainlm)

Retrain

Results

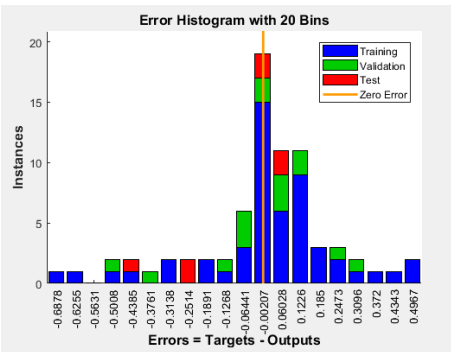
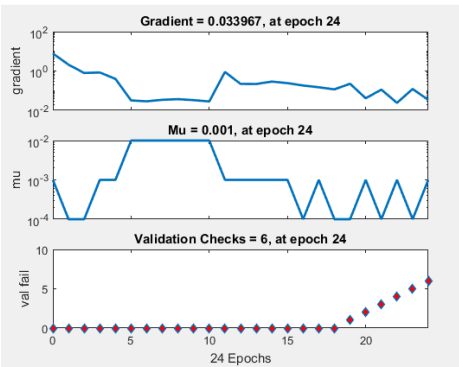
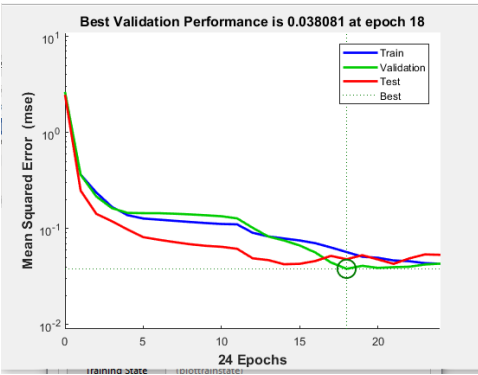
	Samples	MSE	R
Training:	52	5.69200e-2	8.17088e-1
Validation:	15	3.80814e-2	9.00022e-1
Testing:	7	4.79190e-2	8.66667e-1

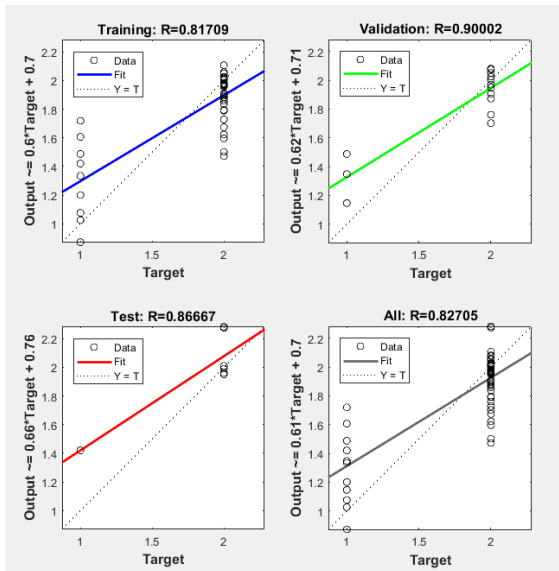
Plot Fit
Plot Error Histogram

Plot Regression

Notes

- Training multiple times will generate different results due to different initial conditions and sampling.
- Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.
- Regression R Values measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship.





Pruebas 65% entrenamiento, 20% validación, 15% pruebas con 74 muestras

Neural Fitting (nftool)

Validation and Test Data

Set aside some samples for validation and testing.

Select Percentages		Explanation	
Randomly divide up the 74 samples:			
Training:	65%	48 samples	Three Kinds of Samples: Training: These are presented to the network during training, and the network is adjusted according to its error. Validation: These are used to measure network generalization, and to halt training when generalization stops improving. Testing: These have no effect on training and so provide an independent measure of network performance during and after training.
Validation:	20%	15 samples	
Testing:	15%	11 samples	

Network Architecture

Set the number of neurons in the fitting network's hidden layer.

Hidden Layer

Define a fitting neural network. (fitnet)

Number of Hidden Neurons:

[Restore Defaults](#)

Recommendation

Return to this panel and change the number of neurons if the network does not perform well after training.

Neural Network

Train Network

Train the network to fit the inputs and targets.

Train Network

Choose a training algorithm:

This algorithm typically requires more memory but less time. Training automatically stops when generalization stops improving, as indicated by an increase in the mean square error of the validation samples.

Train using Levenberg-Marquardt. (trainlm)

[Retrain](#)

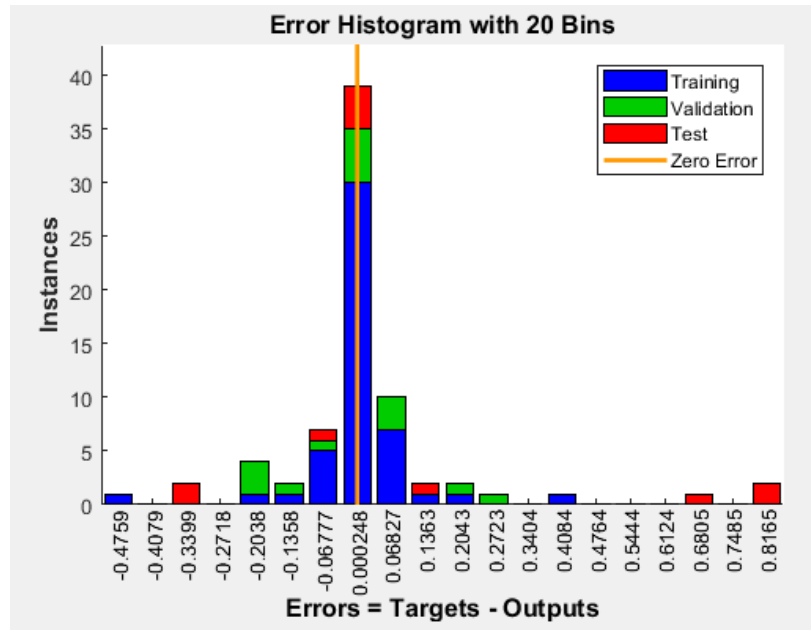
Results

	Samples	MSE	R
Training:	48	1.27586e-2	9.53058e-1
Validation:	15	1.88086e-2	9.40568e-1
Testing:	11	1.83946e-1	8.01760e-1

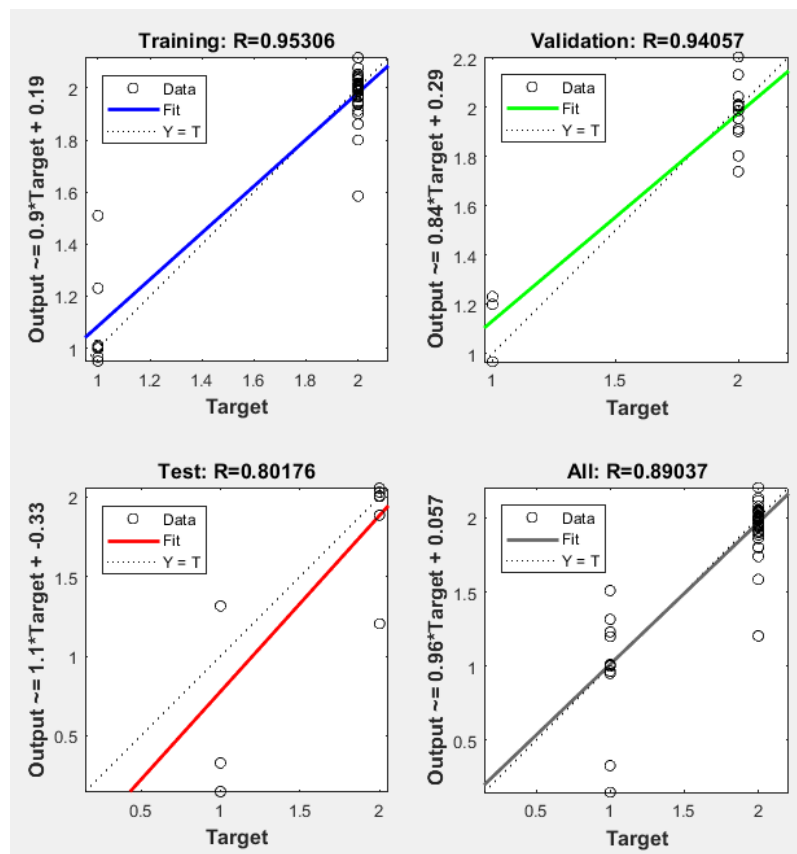
[Plot Fit](#) [Plot Error Histogram](#)
[Plot Regression](#)

Notes

- Training multiple times will generate different results due to different initial conditions and sampling.
- Mean Squared Error is the average squared difference between outputs and targets. Lower values are better. Zero means no error.
- Regression R Values measure the correlation between outputs and targets. An R value of 1 means a close relationship, 0 a random relationship.



Histograma de error



Gráfica de regresión

Matlab function

```

function [Y,Xf,Af] = myNeuralNetworkFunction(X,~,~)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 02-Oct-2020
16:25:28.
%
% [Y] = myNeuralNetworkFunction(X,~,~) takes these arguments:
%
% X = 1xTS cell, 1 inputs over TS timesteps
% Each X{1,ts} = 19xQ matrix, input #1 at timestep ts.
%
% and returns:
% Y = 1xTS cell of 1 outputs over TS timesteps.
% Each Y{1,ts} = 1xQ matrix, output #1 at timestep ts.
%
% where Q is number of samples (or series) and TS is the number of timesteps.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset = [0.003856;25.126965;8.3e-
05;2.517872;0;0;0.000104;0.000165;0.000206;5.5e-05;6.2e-05;4.1e-
05;0;0.000165;0;0;-3.671084;2.732656;2.72937];
x1_step1.gain =
[22.0242486978163;0.000752660216947654;63.9611116441204;0.001968994055674
88;0.000119410114036659;0.000107625248883388;32.3086118604914;32.24610225
23902;16.285451391999;175.947919415853;85.0448611642641;128.09017548354;4
.52190978337791;0.0307555157018749;520.156046814044;0.000119338862700638;
0.310411560719838;10.5815626851774;10.2376150451991];
x1_step1.ymin = -1;

% Layer 1
b1 = [-1.7101538678313501;0.8833013997558905;-
1.0654623054709318;0.2848610089631074;0.19067682679521039;0.1581940211655

```

8892;-0.90861850906276231;1.5472780755428526;-1.4247361759681836;-
1.7121742545393941];

IW1_1 = [0.013794847753043692 0.479475377971192 -0.043388814419406486
0.12763944812648073 0.10651885061945647 -0.36159675225500626
0.27243368295637438 -0.60721640540508548 0.2609391921591882
0.17980402877035745 -0.39904814324892152 -0.12154969384119406
0.40169366352356312 0.36759514437076629 0.44485626070559492 -
0.31793958952652945 -0.65741888970428064 0.26992554354791354 -
0.45697227099770171;-0.46272958251833579 -0.56411371288701229
0.71437373566211804 0.41717256030014582 0.26328425538006089
0.28466772381433703 0.19295839663840106 0.17194337845859814 -
0.26995473970501238 0.28798793727580707 0.25259982863883534
0.47367108733994961 0.21804711893180229 0.96490527669065473
0.55236587810552462 0.0045280967564247888 0.17713747010932193 -
0.25636717851739305 -0.27994693979400581;-1.5032811563815078
0.93767245472231964 -1.0222116650464053 -0.023008105384592123 -
0.1019198513065747 -0.014261140900194449 -0.44345580238195992 -
0.21431263321886221 0.63339787944467496 -1.472624706005824
0.04208918813859882 -0.58261447335477923 -0.74458471866055354
0.28735504602599798 0.3629379707259584 -0.50303028063444544 -
1.7816817934258904 -1.5432681660972416 -
0.65570705457603851;0.72625200155756842 0.19393584940709138
0.092325357168985531 -0.17635622001477033 0.89458713730324124
0.6003469438748692 0.041785935264687907 -0.38882146480254659 -
0.43495196037259459 0.21187709623729062 -0.77491728098228729
0.09778910626297907 -0.92380302360035416 -0.21692827148184129 -
0.85140414970632439 0.83164579119472781 1.1078528226188595
1.9736018088101401 1.8509094686611083;-0.13237428837277412
1.3850169542287567 -0.096912825553089929 0.33231013148740507 -
0.26082964765537153 0.014337324491241333 -0.1411003825722876 -
0.03478151430864708 0.14839248828468957 -0.73190689670009668
0.020086228840938771 0.3671699909134144 -0.16083343116825369 -
0.037843292322786715 0.11124664566209209 -0.37524748389612894 -
1.4968152432955308 4.0574761365336736 2.7480644037621453;-
0.94585182947781343 -0.54174842913297372 0.19353449046102653 -
0.12286079558298064 0.23866380352608652 -0.033993637963172681 -
0.6258732267139433 -0.092389932459068458 0.22067373759860362 -
0.20110266080630515 0.1859847843695685 -0.32774010457001163

```

0.40173789540813387    -0.27334846643303057    -0.31275924275235645    -
0.37098031052556441        3.2180299720021681        -0.14142127002859883
7.7177885077412354;-1.7729321521701558        -1.7760295972352231
0.63436226687276998        -0.041606086020237763        0.6315807296857785
0.66798387990013031        -0.33256029535989573        0.14625598346971258
0.3267300346010078        0.28578986897149283        -0.077675662618256924
0.94600638525951564        1.319604287142063        0.44326115771670671
1.188082983316334        -0.26103767926959992        -4.3766247632283397    -
0.2590074527896094        1.0066595125579514;-1.8384506895440778
2.362263981651433    0.088718198920537561    -0.080266627822589962    -
0.91474739692494023        -0.092846139186339896        -0.74023826612080479
0.10064055994380883        -0.57166207330507168        -0.79550176050703802    -
0.40915121337219046        0.40802833869001942        -0.31281496561405747    -
0.31233773231851308        -0.59166591180285366        -0.73483679758012133
6.9773959576213445        7.5250192818127655        -6.9482024342838056;-
1.8158872171193137    2.5010238240539255    -0.21446999060652594    -
0.034244532124751911        -0.59676033285612751        -0.81893553755034387    -
0.028194449381171157        -0.56148888756711979        -1.1225467636533544    -
0.90901588501806763        -0.44709453057212367        0.028530679042960572
0.00069523474169054696        -0.65591023309087559        0.17362013736864582    -
0.97205511100374686        0.78661901343277685        -3.7478317639175454    -
2.9326976637918936;-0.21892378211132754        -0.54508198669455243
0.46605821317630386        0.07288114978714294        0.17324408051013765
0.018328635875664058        0.018041985121370757        -0.25527212532018106
0.0084691709052113866        0.38358065842654743        0.25998078755172122
0.45201952053099143        -0.026325753459633117        0.40426246065071703
0.29231617194802534        0.63316788528686307        0.0068558687579522949
0.09640801873170271 0.24551228203412481];

```

```
% Layer 2
```

```

b2 = 0.8621508231042796;
LW2_1 = [-0.84853609558509635  0.14996593555651461  -3.1284331625649977  -
2.1692085340093503        -4.2656308945117836        5.1435038721949553
3.9312687803888191        2.8495018238939784        1.7569178794361546    -
0.62071065252317237];

```

```
% Output 1
```

```
y1_step1.ymin = -1;
```



```

y1_step1.gain = 2;
y1_step1.xoffset = 1;

% ===== SIMULATION =====

% Format Input Arguments
isCellX = iscell(X);
if ~isCellX
    X = {X};
end

% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
    Q = size(X{1},2); % samples/series
else
    Q = 0;
end

% Allocate Outputs
Y = cell(1,TS);

% Time loop
for ts=1:TS

    % Input 1
    Xp1 = mapminmax_apply(X{1,ts},x1_step1);

    % Layer 1
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);

    % Layer 2
    a2 = repmat(b2,1,Q) + LW2_1*a1;

    % Output 1
    Y{1,ts} = mapminmax_reverse(a2,y1_step1);

```

```

end

% Final Delay States
Xf = cell(1,0);
Af = cell(2,0);

% Format Output Arguments
if ~isCellX
    Y = cell2mat(Y);
end
end

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
x = bsxfun(@minus,y,settings.ymin);
x = bsxfun(@rdivide,x,settings.gain);
x = bsxfun(@plus,x,settings.xoffset);
end

```

Matlab matrix only function

```

function [y1] = myNeuralNetworkFunction(x1)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 02-Oct-2020
16:25:43.
%
% [y1] = myNeuralNetworkFunction(x1) takes these arguments:
%   x = 19xQ matrix, input #1
% and returns:
%   y = 1xQ matrix, output #1
% where Q is the number of samples.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1.xoffset          =          [0.003856;25.126965;8.3e-
05;2.517872;0;0;0.000104;0.000165;0.000206;5.5e-05;6.2e-05;4.1e-
05;0;0.000165;0;0;-3.671084;2.732656;2.72937];
x1_step1.gain              =
[22.0242486978163;0.000752660216947654;63.9611116441204;0.001968994055674
88;0.000119410114036659;0.000107625248883388;32.3086118604914;32.24610225
23902;16.285451391999;175.947919415853;85.0448611642641;128.09017548354;4
.52190978337791;0.0307555157018749;520.156046814044;0.000119338862700638;
0.310411560719838;10.5815626851774;10.2376150451991];
x1_step1.ymin = -1;

% Layer 1
b1          =          [-1.7101538678313501;0.8833013997558905;-
1.0654623054709318;0.2848610089631074;0.19067682679521039;0.1581940211655
8892;-0.90861850906276231;1.5472780755428526;-1.4247361759681836;-
1.7121742545393941];
IW1_1      = [0.013794847753043692   0.479475377971192   -0.043388814419406486
0.12763944812648073           0.10651885061945647           -0.36159675225500626
0.27243368295637438           -0.60721640540508548           0.2609391921591882
0.17980402877035745           -0.39904814324892152           -0.12154969384119406

```

0.40169366352356312	0.36759514437076629	0.44485626070559492	-
0.31793958952652945	-0.65741888970428064	0.26992554354791354	-
0.45697227099770171;	-0.46272958251833579	-0.56411371288701229	
0.71437373566211804	0.41717256030014582	0.26328425538006089	
0.28466772381433703	0.19295839663840106	0.17194337845859814	-
0.26995473970501238	0.28798793727580707	0.25259982863883534	
0.47367108733994961	0.21804711893180229	0.96490527669065473	
0.55236587810552462	0.0045280967564247888	0.17713747010932193	-
0.25636717851739305	-0.27994693979400581;	-1.5032811563815078	
0.93767245472231964	-1.0222116650464053	-0.023008105384592123	-
0.1019198513065747	-0.014261140900194449	-0.44345580238195992	-
0.21431263321886221	0.63339787944467496	-1.472624706005824	
0.04208918813859882	-0.58261447335477923	-0.74458471866055354	
0.28735504602599798	0.3629379707259584	-0.50303028063444544	-
1.7816817934258904	-1.5432681660972416		-
0.65570705457603851;	0.72625200155756842	0.19393584940709138	
0.092325357168985531	-0.17635622001477033	0.89458713730324124	
0.6003469438748692	0.041785935264687907	-0.38882146480254659	-
0.43495196037259459	0.21187709623729062	-0.77491728098228729	
0.09778910626297907	-0.92380302360035416	-0.21692827148184129	-
0.85140414970632439	0.83164579119472781	1.1078528226188595	
1.9736018088101401	1.8509094686611083;	-0.13237428837277412	
1.3850169542287567	-0.096912825553089929	0.33231013148740507	-
0.26082964765537153	0.014337324491241333	-0.1411003825722876	-
0.03478151430864708	0.14839248828468957	-0.73190689670009668	
0.020086228840938771	0.3671699909134144	-0.16083343116825369	-
0.037843292322786715	0.11124664566209209	-0.37524748389612894	-
1.4968152432955308	4.0574761365336736	2.7480644037621453;	-
0.94585182947781343	-0.54174842913297372	0.19353449046102653	-
0.12286079558298064	0.23866380352608652	-0.033993637963172681	-
0.6258732267139433	-0.092389932459068458	0.22067373759860362	-
0.20110266080630515	0.1859847843695685	-0.32774010457001163	
0.40173789540813387	-0.27334846643303057	-0.31275924275235645	-
0.37098031052556441	3.2180299720021681	-0.14142127002859883	
7.7177885077412354;	-1.7729321521701558	-1.7760295972352231	
0.63436226687276998	-0.041606086020237763	0.6315807296857785	
0.66798387990013031	-0.33256029535989573	0.14625598346971258	
0.3267300346010078	0.28578986897149283	-0.077675662618256924	

```

0.94600638525951564      1.319604287142063      0.44326115771670671
1.188082983316334      -0.26103767926959992      -4.3766247632283397      -
0.2590074527896094      1.0066595125579514;-1.8384506895440778
2.362263981651433      0.088718198920537561      -0.080266627822589962      -
0.91474739692494023      -0.092846139186339896      -0.74023826612080479
0.10064055994380883      -0.57166207330507168      -0.79550176050703802      -
0.40915121337219046      0.40802833869001942      -0.31281496561405747      -
0.31233773231851308      -0.59166591180285366      -0.73483679758012133
6.9773959576213445      7.5250192818127655      -6.9482024342838056;-
1.8158872171193137      2.5010238240539255      -0.21446999060652594      -
0.034244532124751911      -0.59676033285612751      -0.81893553755034387      -
0.028194449381171157      -0.56148888756711979      -1.1225467636533544      -
0.90901588501806763      -0.44709453057212367      0.028530679042960572
0.00069523474169054696      -0.65591023309087559      0.17362013736864582      -
0.97205511100374686      0.78661901343277685      -3.7478317639175454      -
2.9326976637918936;-0.21892378211132754      -0.54508198669455243
0.46605821317630386      0.07288114978714294      0.17324408051013765
0.018328635875664058      0.018041985121370757      -0.25527212532018106
0.0084691709052113866      0.38358065842654743      0.25998078755172122
0.45201952053099143      -0.026325753459633117      0.40426246065071703
0.29231617194802534      0.63316788528686307      0.0068558687579522949
0.09640801873170271 0.24551228203412481];

```

```
% Layer 2
```

```

b2 = 0.8621508231042796;
LW2_1 = [-0.84853609558509635  0.14996593555651461  -3.1284331625649977  -
2.1692085340093503      -4.2656308945117836      5.1435038721949553
3.9312687803888191      2.8495018238939784      1.7569178794361546      -
0.62071065252317237];

```

```
% Output 1
```

```

y1_step1.ymin = -1;
y1_step1.gain = 2;
y1_step1.xoffset = 1;

```

```
% ===== SIMULATION =====
```

```

% Dimensions
Q = size(x1,2); % samples

% Input 1
xp1 = mapminmax_apply(x1,x1_step1);

% Layer 1
a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*xp1);

% Layer 2
a2 = repmat(b2,1,Q) + LW2_1*a1;

% Output 1
y1 = mapminmax_reverse(a2,y1_step1);
end

% ===== MODULE FUNCTIONS =====

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings)
y = bsxfun(@minus,x,settings.xoffset);
y = bsxfun(@times,y,settings.gain);
y = bsxfun(@plus,y,settings.ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n,~)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings)
x = bsxfun(@minus,y,settings.ymin);
x = bsxfun(@rdivide,x,settings.gain);
x = bsxfun(@plus,x,settings.xoffset);
end

```

Script

```
% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by Neural Fitting app
% Created 02-Oct-2020 16:27:26
%
% This script assumes these variables are defined:
%
%   EMGFullldata - input data.
%   EMGTargetFullldata - target data.

x = EMGFullldata;
t = EMGTargetFullldata;

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize,trainFcn);

% Setup Division of Data for Training, Validation, Testing
net.divideParam.trainRatio = 65/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 20/100;

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
```

```
performance = perform(net,t,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)
```

Advanced script

```
% Solve an Input-Output Fitting problem with a Neural Network
% Script generated by Neural Fitting app
% Created 02-Oct-2020 16:28:01
%
% This script assumes these variables are defined:
%
%   EMGFullldata - input data.
%   EMGTargetFullldata - target data.

x = EMGFullldata;
t = EMGTargetFullldata;

% Choose a Training Function
% For a list of all training functions type: help nntrain
% 'trainlm' is usually fastest.
% 'trainbr' takes longer but may be better for challenging problems.
% 'trainscg' uses less memory. Suitable in low memory situations.
trainFcn = 'trainlm'; % Levenberg-Marquardt backpropagation.

% Create a Fitting Network
```



```

hiddenLayerSize = 10;
net = fitnet(hiddenLayerSize,trainFcn);

% Choose Input and Output Pre/Post-Processing Functions
% For a list of all processing functions type: help nnprocess
net.input.processFcns = {'removeconstantrows','mapminmax'};
net.output.processFcns = {'removeconstantrows','mapminmax'};

% Setup Division of Data for Training, Validation, Testing
% For a list of all data division functions type: help nndivide
net.divideFcn = 'dividerand'; % Divide data randomly
net.divideMode = 'sample'; % Divide up every sample
net.divideParam.trainRatio = 65/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 20/100;

% Choose a Performance Function
% For a list of all performance functions type: help nnperformance
net.performFcn = 'mse'; % Mean Squared Error

% Choose Plot Functions
% For a list of all plot functions type: help nnplot
net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
    'plotregression','plotfit'};

% Train the Network
[net,tr] = train(net,x,t);

% Test the Network
y = net(x);
e = gsubtract(t,y);
performance = perform(net,t,y)

% Recalculate Training, Validation and Test Performance
trainTargets = t .* tr.trainMask{1};
valTargets = t .* tr.valMask{1};
testTargets = t .* tr.testMask{1};

```

```
trainPerformance = perform(net,trainTargets,y)
valPerformance = perform(net,valTargets,y)
testPerformance = perform(net,testTargets,y)

% View the Network
view(net)

% Plots
% Uncomment these lines to enable various plots.
%figure, plotperform(tr)
%figure, plottrainstate(tr)
%figure, ploterrhist(e)
%figure, plotregression(t,y)
%figure, plotfit(net,x,t)

% Deployment
% Change the (false) values to (true) to enable the following code blocks.
% See the help for each generation function for more information.
if (false)
    % Generate MATLAB function for neural network for application
    % deployment in MATLAB scripts or with MATLAB Compiler and Builder
    % tools, or simply to examine the calculations your trained neural
    % network performs.
    genFunction(net,'myNeuralNetworkFunction');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a matrix-only MATLAB function for neural network code
    % generation with MATLAB Coder tools.
    genFunction(net,'myNeuralNetworkFunction','MatrixOnly','yes');
    y = myNeuralNetworkFunction(x);
end
if (false)
    % Generate a Simulink diagram for simulation or deployment with.
    % Simulink Coder tools.
    gensim(net);
end
```

Anexo 6. Señales EMG

Adquisición propia

Las señales mostradas a continuación, son solo una brevísima muestra de lo que se tiene almacenado. Si usted amigo lector desea experimentar, puede solicitar las señales a mi correo personal tityslan@gmail.com y con todo gusto se las enviaré.

Tiempo / Amplitud	s1		s2		s3	s4	s5	
0.0001	626	1.11328125	260	-2.4609375	-5.02403259	-5.04906282	689	1.72851563
0.0002	625	1.10351563	259	-2.47070313	-5.02412796	-5.04906375	683	1.66992188
0.0003	625	1.10351563	259	-2.47070313	-5.02412796	-5.04906375	674	1.58203125
0.0004	624	1.09375	258	-2.48046875	-5.02422333	-5.04906468	663	1.47460938
0.0005	624	1.09375	257	-2.49023438	-5.0243187	-5.04906561	651	1.35742188
0.0006	624	1.09375	257	-2.49023438	-5.0243187	-5.04906561	636	1.2109375
0.0007	623	1.08398438	256	-2.5	-5.02441406	-5.04906654	620	1.0546875
0.0008	623	1.08398438	256	-2.5	-5.02441406	-5.04906654	604	0.8984375
0.0009	623	1.08398438	255	-2.50976563	-5.02450943	-5.04906747	585	0.71289063
0.001	622	1.07421875	255	-2.50976563	-5.02450943	-5.04906747	567	0.53710938
0.0011	623	1.08398438	254	-2.51953125	-5.0246048	-5.04906841	551	0.38085938
0.0012	622	1.07421875	254	-2.51953125	-5.0246048	-5.04906841	535	0.22460938
0.0013	621	1.06445313	253	-2.52929688	-5.02470016	-5.04906934	519	0.06835938
0.0014	621	1.06445313	253	-2.52929688	-5.02470016	-5.04906934	505	-0.06835938
0.0015	621	1.06445313	253	-2.52929688	-5.02470016	-5.04906934	489	-0.22460938
0.0016	620	1.0546875	252	-2.5390625	-5.02479553	-5.04907027	476	-0.3515625
0.0017	620	1.0546875	252	-2.5390625	-5.02479553	-5.04907027	464	-0.46875
0.0018	619	1.04492188	252	-2.5390625	-5.02479553	-5.04907027	449	-0.61523438
0.0019	619	1.04492188	252	-2.5390625	-5.02479553	-5.04907027	437	-0.73242188
0.002	619	1.04492188	251	-2.54882813	-5.0248909	-5.0490712	426	-0.83984375
0.0021	618	1.03515625	251	-2.54882813	-5.0248909	-5.0490712	413	-0.96679688
0.0022	618	1.03515625	251	-2.54882813	-5.0248909	-5.0490712	401	-1.08398438
0.0023	618	1.03515625	251	-2.54882813	-5.0248909	-5.0490712	391	-1.18164063
0.0024	618	1.03515625	250	-2.55859375	-5.02498627	-5.04907213	379	-1.29882813
0.0025	617	1.02539063	250	-2.55859375	-5.02498627	-5.04907213	369	-1.39648438
0.0026	616	1.015625	250	-2.55859375	-5.02498627	-5.04907213	359	-1.49414063
0.0027	616	1.015625	250	-2.55859375	-5.02498627	-5.04907213	349	-1.59179688
0.0028	615	1.00585938	250	-2.55859375	-5.02498627	-5.04907213	340	-1.6796875
0.0029	615	1.00585938	250	-2.55859375	-5.02498627	-5.04907213	331	-1.76757813
0.003	615	1.00585938	250	-2.55859375	-5.02498627	-5.04907213	322	-1.85546875
0.0031	614	0.99609375	250	-2.55859375	-5.02498627	-5.04907213	314	-1.93359375

0.0032	614	0.99609375	250	-2.55859375	-5.02498627	-5.04907213	305	-2.02148438
0.0033	613	0.98632813	250	-2.55859375	-5.02498627	-5.04907213	297	-2.09960938
0.0034	613	0.98632813	250	-2.55859375	-5.02498627	-5.04907213	290	-2.16796875
0.0035	613	0.98632813	250	-2.55859375	-5.02498627	-5.04907213	282	-2.24609375
0.0036	613	0.98632813	250	-2.55859375	-5.02498627	-5.04907213	275	-2.31445313
0.0037	612	0.9765625	250	-2.55859375	-5.02498627	-5.04907213	268	-2.3828125
0.0038	611	0.96679688	250	-2.55859375	-5.02498627	-5.04907213	261	-2.45117188
0.0039	611	0.96679688	250	-2.55859375	-5.02498627	-5.04907213	254	-2.51953125
0.004	610	0.95703125	250	-2.55859375	-5.02498627	-5.04907213	247	-2.58789063
0.0041	610	0.95703125	250	-2.55859375	-5.02498627	-5.04907213	241	-2.64648438
0.0042	609	0.94726563	250	-2.55859375	-5.02498627	-5.04907213	236	-2.6953125
0.0043	609	0.94726563	251	-2.54882813	-5.0248909	-5.0490712	229	-2.76367188
0.0044	609	0.94726563	251	-2.54882813	-5.0248909	-5.0490712	223	-2.82226563
0.0045	608	0.9375	250	-2.55859375	-5.02498627	-5.04907213	218	-2.87109375
0.0046	608	0.9375	250	-2.55859375	-5.02498627	-5.04907213	212	-2.9296875
0.0047	607	0.92773438	250	-2.55859375	-5.02498627	-5.04907213	207	-2.97851563
0.0048	607	0.92773438	250	-2.55859375	-5.02498627	-5.04907213	202	-3.02734375
0.0049	606	0.91796875	250	-2.55859375	-5.02498627	-5.04907213	197	-3.07617188
0.005	606	0.91796875	250	-2.55859375	-5.02498627	-5.04907213	192	-3.125
0.0051	605	0.90820313	250	-2.55859375	-5.02498627	-5.04907213	187	-3.17382813
0.0052	604	0.8984375	250	-2.55859375	-5.02498627	-5.04907213	182	-3.22265625
0.0053	604	0.8984375	250	-2.55859375	-5.02498627	-5.04907213	178	-3.26171875
0.0054	603	0.88867188	250	-2.55859375	-5.02498627	-5.04907213	173	-3.31054688
0.0055	603	0.88867188	249	-2.56835938	-5.02508163	-5.04907306	169	-3.34960938
0.0056	602	0.87890625	250	-2.55859375	-5.02498627	-5.04907213	165	-3.38867188
0.0057	601	0.86914063	249	-2.56835938	-5.02508163	-5.04907306	161	-3.42773438
0.0058	601	0.86914063	249	-2.56835938	-5.02508163	-5.04907306	157	-3.46679688
0.0059	600	0.859375	249	-2.56835938	-5.02508163	-5.04907306	153	-3.50585938
0.006	600	0.859375	249	-2.56835938	-5.02508163	-5.04907306	150	-3.53515625
0.0061	599	0.84960938	249	-2.56835938	-5.02508163	-5.04907306	146	-3.57421875
0.0062	599	0.84960938	248	-2.578125	-5.025177	-5.04907399	143	-3.60351563
0.0063	597	0.83007813	248	-2.578125	-5.025177	-5.04907399	139	-3.64257813
0.0064	597	0.83007813	248	-2.578125	-5.025177	-5.04907399	136	-3.671875
0.0065	597	0.83007813	248	-2.578125	-5.025177	-5.04907399	132	-3.7109375
0.0066	596	0.8203125	247	-2.58789063	-5.02527237	-5.04907493	129	-3.74023438
0.0067	596	0.8203125	247	-2.58789063	-5.02527237	-5.04907493	126	-3.76953125
0.0068	595	0.81054688	247	-2.58789063	-5.02527237	-5.04907493	123	-3.79882813
0.0069	594	0.80078125	247	-2.58789063	-5.02527237	-5.04907493	120	-3.828125
0.007	594	0.80078125	247	-2.58789063	-5.02527237	-5.04907493	117	-3.85742188
0.0071	593	0.79101563	247	-2.58789063	-5.02527237	-5.04907493	114	-3.88671875
0.0072	593	0.79101563	247	-2.58789063	-5.02527237	-5.04907493	111	-3.91601563

0.0073	593	0.79101563	247	-2.58789063	-5.02527237	-5.04907493	109	-3.93554688
0.0074	592	0.78125	247	-2.58789063	-5.02527237	-5.04907493	106	-3.96484375
0.0075	591	0.77148438	246	-2.59765625	-5.02536774	-5.04907586	104	-3.984375
0.0076	591	0.77148438	247	-2.58789063	-5.02527237	-5.04907493	101	-4.01367188
0.0077	591	0.77148438	247	-2.58789063	-5.02527237	-5.04907493	99	-4.03320313
0.0078	590	0.76171875	247	-2.58789063	-5.02527237	-5.04907493	96	-4.0625
0.0079	590	0.76171875	247	-2.58789063	-5.02527237	-5.04907493	94	-4.08203125
0.008	589	0.75195313	247	-2.58789063	-5.02527237	-5.04907493	92	-4.1015625
0.0081	589	0.75195313	247	-2.58789063	-5.02527237	-5.04907493	90	-4.12109375
0.0082	589	0.75195313	247	-2.58789063	-5.02527237	-5.04907493	88	-4.140625
0.0083	589	0.75195313	247	-2.58789063	-5.02527237	-5.04907493	86	-4.16015625
0.0084	588	0.7421875	247	-2.58789063	-5.02527237	-5.04907493	83	-4.18945313
0.0085	588	0.7421875	247	-2.58789063	-5.02527237	-5.04907493	81	-4.20898438
0.0086	588	0.7421875	247	-2.58789063	-5.02527237	-5.04907493	80	-4.21875
0.0087	587	0.73242188	247	-2.58789063	-5.02527237	-5.04907493	78	-4.23828125
0.0088	586	0.72265625	247	-2.58789063	-5.02527237	-5.04907493	76	-4.2578125
0.0089	587	0.73242188	248	-2.578125	-5.025177	-5.04907399	74	-4.27734375
0.009	586	0.72265625	248	-2.578125	-5.025177	-5.04907399	73	-4.28710938
0.0091	586	0.72265625	248	-2.578125	-5.025177	-5.04907399	71	-4.30664063
0.0092	586	0.72265625	248	-2.578125	-5.025177	-5.04907399	69	-4.32617188
0.0093	586	0.72265625	248	-2.578125	-5.025177	-5.04907399	67	-4.34570313
0.0094	585	0.71289063	249	-2.56835938	-5.02508163	-5.04907306	66	-4.35546875
0.0095	585	0.71289063	249	-2.56835938	-5.02508163	-5.04907306	65	-4.36523438
0.0096	585	0.71289063	249	-2.56835938	-5.02508163	-5.04907306	63	-4.38476563
0.0097	585	0.71289063	249	-2.56835938	-5.02508163	-5.04907306	62	-4.39453125
0.0098	584	0.703125	249	-2.56835938	-5.02508163	-5.04907306	61	-4.40429688
0.0099	584	0.703125	250	-2.55859375	-5.02498627	-5.04907213	59	-4.42382813
0.01	585	0.71289063	250	-2.55859375	-5.02498627	-5.04907213	58	-4.43359375
0.0101	584	0.703125	250	-2.55859375	-5.02498627	-5.04907213	56	-4.453125
0.0102	584	0.703125	250	-2.55859375	-5.02498627	-5.04907213	55	-4.46289063
0.0103	584	0.703125	250	-2.55859375	-5.02498627	-5.04907213	54	-4.47265625
0.0104	584	0.703125	251	-2.54882813	-5.0248909	-5.0490712	52	-4.4921875
0.0105	584	0.703125	251	-2.54882813	-5.0248909	-5.0490712	51	-4.50195313
0.0106	584	0.703125	251	-2.54882813	-5.0248909	-5.0490712	50	-4.51171875
0.0107	584	0.703125	251	-2.54882813	-5.0248909	-5.0490712	49	-4.52148438
0.0108	584	0.703125	252	-2.5390625	-5.02479553	-5.04907027	48	-4.53125
0.0109	583	0.69335938	252	-2.5390625	-5.02479553	-5.04907027	47	-4.54101563
0.011	583	0.69335938	252	-2.5390625	-5.02479553	-5.04907027	46	-4.55078125
0.0111	584	0.703125	252	-2.5390625	-5.02479553	-5.04907027	45	-4.56054688
0.0112	583	0.69335938	253	-2.52929688	-5.02470016	-5.04906934	44	-4.5703125
0.0113	583	0.69335938	253	-2.52929688	-5.02470016	-5.04906934	44	-4.5703125

0.0114	583	0.69335938	253	-2.52929688	-5.02470016	-5.04906934	46	-4.55078125
0.0115	583	0.69335938	253	-2.52929688	-5.02470016	-5.04906934	50	-4.51171875
0.0116	583	0.69335938	254	-2.51953125	-5.0246048	-5.04906841	56	-4.453125
0.0117	583	0.69335938	254	-2.51953125	-5.0246048	-5.04906841	64	-4.375
0.0118	584	0.703125	254	-2.51953125	-5.0246048	-5.04906841	74	-4.27734375
0.0119	584	0.703125	254	-2.51953125	-5.0246048	-5.04906841	85	-4.16992188
0.012	584	0.703125	254	-2.51953125	-5.0246048	-5.04906841	99	-4.03320313
0.0121	584	0.703125	255	-2.50976563	-5.02450943	-5.04906747	114	-3.88671875
0.0122	584	0.703125	255	-2.50976563	-5.02450943	-5.04906747	130	-3.73046875
0.0123	585	0.71289063	255	-2.50976563	-5.02450943	-5.04906747	147	-3.56445313
0.0124	585	0.71289063	255	-2.50976563	-5.02450943	-5.04906747	166	-3.37890625
0.0125	585	0.71289063	256	-2.5	-5.02441406	-5.04906654	187	-3.17382813
0.0126	585	0.71289063	256	-2.5	-5.02441406	-5.04906654	208	-2.96875
0.0127	585	0.71289063	256	-2.5	-5.02441406	-5.04906654	231	-2.74414063
0.0128	586	0.72265625	256	-2.5	-5.02441406	-5.04906654	252	-2.5390625
0.0129	586	0.72265625	256	-2.5	-5.02441406	-5.04906654	273	-2.33398438
0.013	587	0.73242188	256	-2.5	-5.02441406	-5.04906654	294	-2.12890625
0.0131	587	0.73242188	257	-2.49023438	-5.0243187	-5.04906561	313	-1.94335938
0.0132	587	0.73242188	257	-2.49023438	-5.0243187	-5.04906561	332	-1.7578125
0.0133	588	0.7421875	257	-2.49023438	-5.0243187	-5.04906561	351	-1.57226563
0.0134	588	0.7421875	257	-2.49023438	-5.0243187	-5.04906561	368	-1.40625
0.0135	589	0.75195313	257	-2.49023438	-5.0243187	-5.04906561	386	-1.23046875
0.0136	590	0.76171875	258	-2.48046875	-5.02422333	-5.04906468	403	-1.06445313
0.0137	590	0.76171875	258	-2.48046875	-5.02422333	-5.04906468	419	-0.90820313
0.0138	590	0.76171875	258	-2.48046875	-5.02422333	-5.04906468	435	-0.75195313
0.0139	591	0.77148438	258	-2.48046875	-5.02422333	-5.04906468	450	-0.60546875
0.014	591	0.77148438	258	-2.48046875	-5.02422333	-5.04906468	465	-0.45898438
0.0141	592	0.78125	258	-2.48046875	-5.02422333	-5.04906468	480	-0.3125
0.0142	592	0.78125	257	-2.49023438	-5.0243187	-5.04906561	495	-0.16601563
0.0143	593	0.79101563	257	-2.49023438	-5.0243187	-5.04906561	508	-0.0390625
0.0144	593	0.79101563	257	-2.49023438	-5.0243187	-5.04906561	522	0.09765625
0.0145	594	0.80078125	257	-2.49023438	-5.0243187	-5.04906561	535	0.22460938
0.0146	594	0.80078125	257	-2.49023438	-5.0243187	-5.04906561	547	0.34179688
0.0147	595	0.81054688	256	-2.5	-5.02441406	-5.04906654	560	0.46875
0.0148	595	0.81054688	256	-2.5	-5.02441406	-5.04906654	573	0.59570313
0.0149	596	0.8203125	256	-2.5	-5.02441406	-5.04906654	584	0.703125
0.015	596	0.8203125	256	-2.5	-5.02441406	-5.04906654	595	0.81054688
0.0151	597	0.83007813	255	-2.50976563	-5.02450943	-5.04906747	607	0.92773438
0.0152	597	0.83007813	255	-2.50976563	-5.02450943	-5.04906747	617	1.02539063
0.0153	597	0.83007813	254	-2.51953125	-5.0246048	-5.04906841	628	1.1328125
0.0154	598	0.83984375	254	-2.51953125	-5.0246048	-5.04906841	638	1.23046875

0.0155	598	0.83984375	253	-2.52929688	-5.02470016	-5.04906934	647	1.31835938
0.0156	599	0.84960938	253	-2.52929688	-5.02470016	-5.04906934	658	1.42578125
0.0157	599	0.84960938	253	-2.52929688	-5.02470016	-5.04906934	667	1.51367188
0.0158	599	0.84960938	252	-2.5390625	-5.02479553	-5.04907027	676	1.6015625
0.0159	599	0.84960938	252	-2.5390625	-5.02479553	-5.04907027	685	1.68945313
0.016	599	0.84960938	251	-2.54882813	-5.0248909	-5.0490712	694	1.77734375
0.0161	600	0.859375	251	-2.54882813	-5.0248909	-5.0490712	703	1.86523438
0.0162	600	0.859375	251	-2.54882813	-5.0248909	-5.0490712	710	1.93359375
0.0163	599	0.84960938	250	-2.55859375	-5.02498627	-5.04907213	718	2.01171875
0.0164	600	0.859375	250	-2.55859375	-5.02498627	-5.04907213	727	2.09960938
0.0165	600	0.859375	249	-2.56835938	-5.02508163	-5.04907306	733	2.15820313
0.0166	600	0.859375	249	-2.56835938	-5.02508163	-5.04907306	740	2.2265625
0.0167	600	0.859375	249	-2.56835938	-5.02508163	-5.04907306	745	2.27539063
0.0168	600	0.859375	249	-2.56835938	-5.02508163	-5.04907306	746	2.28515625
0.0169	600	0.859375	248	-2.578125	-5.025177	-5.04907399	746	2.28515625
0.017	600	0.859375	248	-2.578125	-5.025177	-5.04907399	747	2.29492188
0.0171	600	0.859375	248	-2.578125	-5.025177	-5.04907399	740	2.2265625
0.0172	600	0.859375	247	-2.58789063	-5.02527237	-5.04907493	734	2.16796875
0.0173	600	0.859375	247	-2.58789063	-5.02527237	-5.04907493	727	2.09960938
0.0174	600	0.859375	247	-2.58789063	-5.02527237	-5.04907493	715	1.98242188
0.0175	600	0.859375	247	-2.58789063	-5.02527237	-5.04907493	703	1.86523438
0.0176	600	0.859375	247	-2.58789063	-5.02527237	-5.04907493	689	1.72851563
0.0177	600	0.859375	246	-2.59765625	-5.02536774	-5.04907586	674	1.58203125
0.0178	600	0.859375	246	-2.59765625	-5.02536774	-5.04907586	656	1.40625
0.0179	600	0.859375	246	-2.59765625	-5.02536774	-5.04907586	638	1.23046875
0.018	600	0.859375	246	-2.59765625	-5.02536774	-5.04907586	618	1.03515625
0.0181	600	0.859375	245	-2.60742188	-5.0254631	-5.04907679	600	0.859375
0.0182	600	0.859375	245	-2.60742188	-5.0254631	-5.04907679	582	0.68359375
0.0183	600	0.859375	245	-2.60742188	-5.0254631	-5.04907679	565	0.51757813
0.0184	600	0.859375	245	-2.60742188	-5.0254631	-5.04907679	548	0.3515625
0.0185	600	0.859375	244	-2.6171875	-5.02555847	-5.04907772	533	0.20507813
0.0186	600	0.859375	244	-2.6171875	-5.02555847	-5.04907772	518	0.05859375
0.0187	599	0.84960938	244	-2.6171875	-5.02555847	-5.04907772	503	-0.08789063
0.0188	599	0.84960938	244	-2.6171875	-5.02555847	-5.04907772	489	-0.22460938
0.0189	599	0.84960938	244	-2.6171875	-5.02555847	-5.04907772	476	-0.3515625
0.019	599	0.84960938	243	-2.62695313	-5.02565384	-5.04907865	462	-0.48828125
0.0191	599	0.84960938	243	-2.62695313	-5.02565384	-5.04907865	450	-0.60546875
0.0192	599	0.84960938	243	-2.62695313	-5.02565384	-5.04907865	438	-0.72265625
0.0193	599	0.84960938	243	-2.62695313	-5.02565384	-5.04907865	425	-0.84960938
0.0194	598	0.83984375	242	-2.63671875	-5.02574921	-5.04907958	413	-0.96679688
0.0195	598	0.83984375	242	-2.63671875	-5.02574921	-5.04907958	403	-1.06445313

0.0196	598	0.83984375	242	-2.63671875	-5.02574921	-5.04907958	391	-1.18164063
0.0197	598	0.83984375	241	-2.64648438	-5.02584457	-5.04908051	381	-1.27929688
0.0198	597	0.83007813	241	-2.64648438	-5.02584457	-5.04908051	372	-1.3671875
0.0199	597	0.83007813	241	-2.64648438	-5.02584457	-5.04908051	361	-1.47460938
0.02	597	0.83007813	240	-2.65625	-5.02593994	-5.04908144	351	-1.57226563
0.0201	597	0.83007813	240	-2.65625	-5.02593994	-5.04908144	342	-1.66015625
0.0202	596	0.8203125	240	-2.65625	-5.02593994	-5.04908144	333	-1.74804688
0.0203	596	0.8203125	239	-2.66601563	-5.02603531	-5.04908238	324	-1.8359375
0.0204	596	0.8203125	239	-2.66601563	-5.02603531	-5.04908238	316	-1.9140625
0.0205	596	0.8203125	239	-2.66601563	-5.02603531	-5.04908238	307	-2.00195313
0.0206	595	0.81054688	239	-2.66601563	-5.02603531	-5.04908238	300	-2.0703125
0.0207	595	0.81054688	238	-2.67578125	-5.02613068	-5.04908331	292	-2.1484375
0.0208	594	0.80078125	238	-2.67578125	-5.02613068	-5.04908331	284	-2.2265625
0.0209	594	0.80078125	238	-2.67578125	-5.02613068	-5.04908331	277	-2.29492188
0.021	594	0.80078125	238	-2.67578125	-5.02613068	-5.04908331	270	-2.36328125
0.0211	594	0.80078125	238	-2.67578125	-5.02613068	-5.04908331	263	-2.43164063
0.0212	593	0.79101563	237	-2.68554688	-5.02622604	-5.04908424	256	-2.5
0.0213	592	0.78125	237	-2.68554688	-5.02622604	-5.04908424	250	-2.55859375
0.0214	592	0.78125	237	-2.68554688	-5.02622604	-5.04908424	244	-2.6171875
0.0215	592	0.78125	237	-2.68554688	-5.02622604	-5.04908424	237	-2.68554688
0.0216	592	0.78125	237	-2.68554688	-5.02622604	-5.04908424	231	-2.74414063
0.0217	591	0.77148438	237	-2.68554688	-5.02622604	-5.04908424	226	-2.79296875
0.0218	591	0.77148438	237	-2.68554688	-5.02622604	-5.04908424	220	-2.8515625
0.0219	590	0.76171875	237	-2.68554688	-5.02622604	-5.04908424	215	-2.90039063
0.022	590	0.76171875	237	-2.68554688	-5.02622604	-5.04908424	209	-2.95898438
0.0221	590	0.76171875	236	-2.6953125	-5.02632141	-5.04908517	204	-3.0078125
0.0222	589	0.75195313	236	-2.6953125	-5.02632141	-5.04908517	199	-3.05664063
0.0223	589	0.75195313	236	-2.6953125	-5.02632141	-5.04908517	194	-3.10546875
0.0224	589	0.75195313	236	-2.6953125	-5.02632141	-5.04908517	189	-3.15429688
0.0225	588	0.7421875	236	-2.6953125	-5.02632141	-5.04908517	184	-3.203125
0.0226	588	0.7421875	236	-2.6953125	-5.02632141	-5.04908517	180	-3.2421875
0.0227	588	0.7421875	237	-2.68554688	-5.02622604	-5.04908424	175	-3.29101563
0.0228	587	0.73242188	237	-2.68554688	-5.02622604	-5.04908424	171	-3.33007813
0.0229	587	0.73242188	237	-2.68554688	-5.02622604	-5.04908424	167	-3.36914063
0.023	586	0.72265625	237	-2.68554688	-5.02622604	-5.04908424	163	-3.40820313
0.0231	586	0.72265625	237	-2.68554688	-5.02622604	-5.04908424	159	-3.44726563
0.0232	586	0.72265625	238	-2.67578125	-5.02613068	-5.04908331	155	-3.48632813
0.0233	585	0.71289063	238	-2.67578125	-5.02613068	-5.04908331	151	-3.52539063
0.0234	585	0.71289063	238	-2.67578125	-5.02613068	-5.04908331	147	-3.56445313
0.0235	584	0.703125	239	-2.66601563	-5.02603531	-5.04908238	144	-3.59375
0.0236	585	0.71289063	239	-2.66601563	-5.02603531	-5.04908238	140	-3.6328125

0.0237	584	0.703125	239	-2.66601563	-5.02603531	-5.04908238	137	-3.66210938
0.0238	583	0.69335938	240	-2.65625	-5.02593994	-5.04908144	134	-3.69140625
0.0239	583	0.69335938	240	-2.65625	-5.02593994	-5.04908144	131	-3.72070313
0.024	583	0.69335938	240	-2.65625	-5.02593994	-5.04908144	127	-3.75976563
0.0241	583	0.69335938	241	-2.64648438	-5.02584457	-5.04908051	124	-3.7890625
0.0242	582	0.68359375	241	-2.64648438	-5.02584457	-5.04908051	121	-3.81835938
0.0243	582	0.68359375	241	-2.64648438	-5.02584457	-5.04908051	118	-3.84765625
0.0244	581	0.67382813	242	-2.63671875	-5.02574921	-5.04907958	116	-3.8671875
0.0245	581	0.67382813	242	-2.63671875	-5.02574921	-5.04907958	113	-3.89648438
0.0246	581	0.67382813	243	-2.62695313	-5.02565384	-5.04907865	110	-3.92578125
0.0247	581	0.67382813	243	-2.62695313	-5.02565384	-5.04907865	108	-3.9453125
0.0248	580	0.6640625	243	-2.62695313	-5.02565384	-5.04907865	105	-3.97460938
0.0249	580	0.6640625	244	-2.6171875	-5.02555847	-5.04907772	102	-4.00390625
0.025	580	0.6640625	244	-2.6171875	-5.02555847	-5.04907772	100	-4.0234375
0.0251	580	0.6640625	245	-2.60742188	-5.0254631	-5.04907679	98	-4.04296875
0.0252	579	0.65429688	245	-2.60742188	-5.0254631	-5.04907679	95	-4.07226563
0.0253	579	0.65429688	245	-2.60742188	-5.0254631	-5.04907679	93	-4.09179688
0.0254	579	0.65429688	246	-2.59765625	-5.02536774	-5.04907586	91	-4.11132813
0.0255	579	0.65429688	246	-2.59765625	-5.02536774	-5.04907586	89	-4.13085938
0.0256	578	0.64453125	246	-2.59765625	-5.02536774	-5.04907586	87	-4.15039063
0.0257	578	0.64453125	247	-2.58789063	-5.02527237	-5.04907493	85	-4.16992188
0.0258	578	0.64453125	247	-2.58789063	-5.02527237	-5.04907493	83	-4.18945313
0.0259	578	0.64453125	247	-2.58789063	-5.02527237	-5.04907493	81	-4.20898438
0.026	578	0.64453125	248	-2.578125	-5.025177	-5.04907399	79	-4.22851563
0.0261	578	0.64453125	248	-2.578125	-5.025177	-5.04907399	77	-4.24804688
0.0262	577	0.63476563	248	-2.578125	-5.025177	-5.04907399	75	-4.26757813
0.0263	577	0.63476563	248	-2.578125	-5.025177	-5.04907399	74	-4.27734375
0.0264	577	0.63476563	249	-2.56835938	-5.02508163	-5.04907306	72	-4.296875
0.0265	577	0.63476563	249	-2.56835938	-5.02508163	-5.04907306	70	-4.31640625
0.0266	577	0.63476563	249	-2.56835938	-5.02508163	-5.04907306	69	-4.32617188
0.0267	576	0.625	250	-2.55859375	-5.02498627	-5.04907213	67	-4.34570313
0.0268	576	0.625	250	-2.55859375	-5.02498627	-5.04907213	65	-4.36523438
0.0269	576	0.625	250	-2.55859375	-5.02498627	-5.04907213	64	-4.375
0.027	576	0.625	250	-2.55859375	-5.02498627	-5.04907213	63	-4.38476563
0.0271	576	0.625	251	-2.54882813	-5.0248909	-5.0490712	61	-4.40429688
0.0272	576	0.625	251	-2.54882813	-5.0248909	-5.0490712	60	-4.4140625
0.0273	575	0.61523438	251	-2.54882813	-5.0248909	-5.0490712	59	-4.42382813
0.0274	575	0.61523438	252	-2.5390625	-5.02479553	-5.04907027	57	-4.44335938
0.0275	575	0.61523438	252	-2.5390625	-5.02479553	-5.04907027	56	-4.453125
0.0276	575	0.61523438	252	-2.5390625	-5.02479553	-5.04907027	55	-4.46289063
0.0277	575	0.61523438	253	-2.52929688	-5.02470016	-5.04906934	53	-4.48242188

0.0278	575	0.61523438	253	-2.52929688	-5.02470016	-5.04906934	52	-4.4921875
0.0279	575	0.61523438	253	-2.52929688	-5.02470016	-5.04906934	51	-4.50195313
0.028	574	0.60546875	254	-2.51953125	-5.0246048	-5.04906841	50	-4.51171875
0.0281	574	0.60546875	254	-2.51953125	-5.0246048	-5.04906841	49	-4.52148438
0.0282	574	0.60546875	254	-2.51953125	-5.0246048	-5.04906841	48	-4.53125
0.0283	574	0.60546875	255	-2.50976563	-5.02450943	-5.04906747	47	-4.54101563
0.0284	574	0.60546875	255	-2.50976563	-5.02450943	-5.04906747	46	-4.55078125
0.0285	574	0.60546875	255	-2.50976563	-5.02450943	-5.04906747	45	-4.56054688
0.0286	574	0.60546875	256	-2.5	-5.02441406	-5.04906654	44	-4.5703125
0.0287	574	0.60546875	256	-2.5	-5.02441406	-5.04906654	43	-4.58007813
0.0288	574	0.60546875	257	-2.49023438	-5.0243187	-5.04906561	43	-4.58007813
0.0289	574	0.60546875	257	-2.49023438	-5.0243187	-5.04906561	46	-4.55078125
0.029	574	0.60546875	257	-2.49023438	-5.0243187	-5.04906561	50	-4.51171875
0.0291	574	0.60546875	257	-2.49023438	-5.0243187	-5.04906561	57	-4.44335938
0.0292	574	0.60546875	258	-2.48046875	-5.02422333	-5.04906468	66	-4.35546875
0.0293	574	0.60546875	258	-2.48046875	-5.02422333	-5.04906468	76	-4.2578125
0.0294	574	0.60546875	258	-2.48046875	-5.02422333	-5.04906468	88	-4.140625
0.0295	574	0.60546875	259	-2.47070313	-5.02412796	-5.04906375	102	-4.00390625
0.0296	574	0.60546875	259	-2.47070313	-5.02412796	-5.04906375	117	-3.85742188
0.0297	574	0.60546875	259	-2.47070313	-5.02412796	-5.04906375	134	-3.69140625
0.0298	574	0.60546875	259	-2.47070313	-5.02412796	-5.04906375	152	-3.515625
0.0299	574	0.60546875	260	-2.4609375	-5.02403259	-5.04906282	172	-3.3203125
0.03	574	0.60546875	260	-2.4609375	-5.02403259	-5.04906282	192	-3.125
0.0301	574	0.60546875	260	-2.4609375	-5.02403259	-5.04906282	214	-2.91015625
0.0302	574	0.60546875	260	-2.4609375	-5.02403259	-5.04906282	236	-2.6953125
0.0303	574	0.60546875	260	-2.4609375	-5.02403259	-5.04906282	258	-2.48046875
0.0304	574	0.60546875	260	-2.4609375	-5.02403259	-5.04906282	279	-2.27539063
0.0305	574	0.60546875	260	-2.4609375	-5.02403259	-5.04906282	299	-2.08007813
0.0306	575	0.61523438	261	-2.45117188	-5.02393723	-5.04906189	319	-1.88476563
0.0307	574	0.60546875	261	-2.45117188	-5.02393723	-5.04906189	337	-1.70898438
0.0308	574	0.60546875	261	-2.45117188	-5.02393723	-5.04906189	356	-1.5234375
0.0309	575	0.61523438	261	-2.45117188	-5.02393723	-5.04906189	373	-1.35742188
0.031	575	0.61523438	261	-2.45117188	-5.02393723	-5.04906189	390	-1.19140625
0.0311	575	0.61523438	262	-2.44140625	-5.02384186	-5.04906096	408	-1.015625
0.0312	575	0.61523438	262	-2.44140625	-5.02384186	-5.04906096	423	-0.86914063
0.0313	574	0.60546875	262	-2.44140625	-5.02384186	-5.04906096	439	-0.71289063
0.0314	575	0.61523438	262	-2.44140625	-5.02384186	-5.04906096	455	-0.55664063
0.0315	575	0.61523438	262	-2.44140625	-5.02384186	-5.04906096	469	-0.41992188
0.0316	575	0.61523438	262	-2.44140625	-5.02384186	-5.04906096	484	-0.2734375
0.0317	575	0.61523438	262	-2.44140625	-5.02384186	-5.04906096	499	-0.12695313
0.0318	575	0.61523438	262	-2.44140625	-5.02384186	-5.04906096	512	0

0.0319	575	0.61523438	262	-2.44140625	-5.02384186	-5.04906096	525	0.12695313
0.032	576	0.625	262	-2.44140625	-5.02384186	-5.04906096	538	0.25390625
0.0321	576	0.625	262	-2.44140625	-5.02384186	-5.04906096	551	0.38085938
0.0322	576	0.625	262	-2.44140625	-5.02384186	-5.04906096	563	0.49804688
0.0323	576	0.625	262	-2.44140625	-5.02384186	-5.04906096	575	0.61523438
0.0324	576	0.625	262	-2.44140625	-5.02384186	-5.04906096	587	0.73242188
0.0325	576	0.625	262	-2.44140625	-5.02384186	-5.04906096	599	0.84960938
0.0326	576	0.625	262	-2.44140625	-5.02384186	-5.04906096	609	0.94726563
0.0327	576	0.625	262	-2.44140625	-5.02384186	-5.04906096	620	1.0546875
0.0328	577	0.63476563	262	-2.44140625	-5.02384186	-5.04906096	631	1.16210938
0.0329	577	0.63476563	261	-2.45117188	-5.02393723	-5.04906189	640	1.25
0.033	577	0.63476563	261	-2.45117188	-5.02393723	-5.04906189	651	1.35742188
0.0331	577	0.63476563	261	-2.45117188	-5.02393723	-5.04906189	660	1.4453125
0.0332	577	0.63476563	261	-2.45117188	-5.02393723	-5.04906189	669	1.53320313
0.0333	578	0.64453125	261	-2.45117188	-5.02393723	-5.04906189	679	1.63085938
0.0334	578	0.64453125	261	-2.45117188	-5.02393723	-5.04906189	687	1.70898438
0.0335	578	0.64453125	261	-2.45117188	-5.02393723	-5.04906189	696	1.796875
0.0336	578	0.64453125	261	-2.45117188	-5.02393723	-5.04906189	705	1.88476563
0.0337	578	0.64453125	261	-2.45117188	-5.02393723	-5.04906189	712	1.953125
0.0338	578	0.64453125	261	-2.45117188	-5.02393723	-5.04906189	720	2.03125
0.0339	578	0.64453125	261	-2.45117188	-5.02393723	-5.04906189	729	2.11914063
0.034	578	0.64453125	261	-2.45117188	-5.02393723	-5.04906189	735	2.17773438
0.0341	578	0.64453125	261	-2.45117188	-5.02393723	-5.04906189	743	2.25585938
0.0342	577	0.63476563	261	-2.45117188	-5.02393723	-5.04906189	751	2.33398438
0.0343	578	0.64453125	262	-2.44140625	-5.02384186	-5.04906096	756	2.3828125
0.0344	578	0.64453125	262	-2.44140625	-5.02384186	-5.04906096	761	2.43164063
0.0345	577	0.63476563	262	-2.44140625	-5.02384186	-5.04906096	765	2.47070313
0.0346	577	0.63476563	262	-2.44140625	-5.02384186	-5.04906096	764	2.4609375
0.0347	577	0.63476563	263	-2.43164063	-5.02374649	-5.04906002	762	2.44140625
0.0348	577	0.63476563	263	-2.43164063	-5.02374649	-5.04906002	761	2.43164063
0.0349	577	0.63476563	263	-2.43164063	-5.02374649	-5.04906002	752	2.34375