

UNIVERSIDAD RICARDO PALMA
FACULTAD DE INGENIERÍA
PROGRAMA DE TITULACIÓN POR TESIS
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA



**SIMULACIÓN Y EVALUACIÓN DE TÉCNICAS DE CLUSTERING
PARA EL RECONOCIMIENTO DE GESTOS ESTÁTICOS EN LA
TRADUCCIÓN DE LA LENGUA DE SEÑAS PERUANA**

TESIS

**PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO ELECTRÓNICO**

PRESENTADA POR:

Bach. ESPINOZA HOYOS, CARLOS ANIBAL

ASESOR: Dr. Ing. HUAMANÍ NAVARRETE, PEDRO F.

LIMA-PERÚ

2020

DEDICATORIA

A mis padres Anibal y Fanny, quienes con su apoyo, consejos y tolerancia son el pilar de mi desarrollo académico y profesional; a mis hermanas que con su apoyo incondicional son un hombro para descansar en mi trayecto y a mis familiares en Trujillo, abuelos, tíos y primos, a quienes extraño y quienes son también mi inspiración.

Carlos Anibal Espinoza Hoyos

AGRADECIMIENTO

A mi asesor, el Ing. Pedro F. Huamani Navarrete, por su apoyo y guía durante el desarrollo de esta tesis.

A mis maestros, quienes clase a clase en nuestras aulas nos formaban como profesionales y personas de bien.

A mis compañeros y amigos, que hicieron más llevadera la estancia universitaria con la fraternidad característica, y a mi siempre grupo de trabajo y estudio (Alex, David y Waylounge), únicamente agradecerles por toda la amistad y anécdotas durante las jornadas de estudio y elaboraciones de proyectos.

Carlos Anibal Espinoza Hoyos

ÍNDICE

RESUMEN	viii
ABSTRACT.....	ix
INTRODUCCIÓN	9
CAPÍTULO I: PLANTEAMIENTO Y DELIMITACIÓN DEL PROBLEMA.....	11
1.1. Descripción del problema	11
1.2. Formulación del problema	13
1.2.1 Problema General	13
1.2.2 Problemas Específicos	13
1.3. Importancia y Justificación del estudio.....	14
1.4. Objetivos de la investigación	14
1.4.1. Objetivo general.....	14
1.4.2. Objetivos específicos	15
1.5. Delimitación del estudio	15
CAPÍTULO II: MARCO TEÓRICO	16
2.1. Marco histórico	16
2.2. Investigaciones relacionadas con el tema	17
2.3. Base teórica	20
2.3.1. Técnicas de Clustering.....	20
2.3.2. Reconocimiento de Gestos Estáticos	26
2.4. Definición de términos básicos	30
2.4.1. Arquitecturas de Redes Neuronales.....	30
2.4.3. <i>Clusters</i> o Grupos	30
2.4.4. MATLAB: Deep Learning Toolbox	30
2.4.5. Neurona Artificial	30
2.5. Diseño de la Investigación	31

2.5.1.	Variables de Investigación.....	31
2.5.2.	Tipo y Método de investigación	31
2.5.3.	Técnicas e instrumentos de recolección de datos	31
2.5.4.	Procedimiento para la recolección de datos.....	32
CAPÍTULO III: DISEÑO DE INGENIERÍA.....		33
3.1.	Diagrama de bloques general	33
3.2.	Diagrama de Flujo de la programación en software	35
3.3.	Acondicionamiento de datos e implementación computacional	36
3.3.1.	Obtención y acondicionamiento de muestras de entrada.....	37
3.3.2.	Red Neuronal de Regresión Generalizada (GRNN)	47
3.3.3.	Red Neuronal de Aprendizaje por Cuantificación Vectorial (LVQNN) ..	50
CAPÍTULO IV: PRUEBAS Y RESULTADOS		55
4.1.	Redes Neuronales de Regresión Generalizada: GRNN	55
4.1.1.	Red de Señas de Regresión Generalizada.....	55
4.1.2.	Red MPU de Regresión Generalizada	57
4.1.3.	Red Conjugada de Regresión Generalizada.....	58
4.2.	Redes Neuronales de Aprendizaje por Cuantificación Vectorial: LVQNN.....	59
4.2.1.	Red de Señas de Aprendizaje por Cuantificación Vectorial.....	59
4.2.2.	Red MPU de Aprendizaje por Cuantificación Vectorial	61
4.2.3.	Red Conjugada de Aprendizaje por Cuantificación Vectorial.....	62
4.3.	Análisis y comparación de datos.....	64
4.3.1.	Tabla de contingencia	64
4.3.2.	Precisión.....	65
4.3.3.	Recall	67
4.3.4.	Magnitud F.....	68
4.4.	Aplicación de Red Conjugada GR en los datos LVQ	69
4.4.1.	Tabla de contingencia	69

4.4.2. Evaluación de Precisión y Recall.....	70
4.5. Aplicación de ruido a las redes neuronales	71
CONCLUSIONES	76
RECOMENDACIONES.....	80
REFERENCIAS BIBLIOGRÁFICAS	81

ÍNDICE DE FIGURAS

Figura N° 1: Arquitectura de Red Neuronal GR.....	21
Figura N° 2: Estructura de capas de una red Multicapa.	22
Figura N° 3: Arquitectura de Red Neuronal LVQ.....	24
Figura N° 4: Estructura de capas de una red Monocapa.....	24
Figura N° 5: Modelo de tabla de contingencia C x M.....	27
Figura N° 6: Alfabeto dactilológico peruano.....	29
Figura N° 7: Prototipo de guantes electrónicos para la traducción de la LSP	33
Figura N° 8: Diagrama de bloques	34
Figura N° 9: Diagrama de flujo	35
Figura N° 10: Disposición de sensores en el guante electrónico.....	37
Figura N° 11: Disposición de pines en Arduino Nano	38
Figura N° 12: Voltajes con los dedos extendidos.....	39
Figura N° 13: Valores de voltaje de los dedos contraídos.....	40
Figura N° 14: Normalización de señal de entrada mediante el preprocesamiento.	46
Figura N° 15: Diagrama de Red GR para señas.	48
Figura N° 16: Diagrama de Red GR para sensor MPU.	48
Figura N° 17: Diagrama de Red LVQ para señas.....	51
Figura N° 18: Diagrama de Red LVQ para sensor MPU.	51
Figura N° 19: Grafica de entrenamiento Error vs Iteraciones	53
Figura N° 20: Herramienta de entrenamiento del software MatLab.	54
Figura N° 21: Tabla de Contingencia para Red de señas GR.....	56
Figura N° 22: Tabla de contingencias para red MPU GR.	57
Figura N° 23: Tabla de contingencia de la Red Conjugada GR.....	58
Figura N° 24: Tabla de contingencias de la red de señas LVQ.....	59
Figura N° 25: Tabla de contingencia para red MPU LVQ.	61
Figura N° 26: Tabla de contingencia Red Conjugada LVQ.....	62
Figura N° 27: Tabla de contingencia Red Conjugada GR en LVQ.....	70

ÍNDICE DE TABLAS

Tabla 1: Lectura en volteos de sensores Flex	40
Tabla 2: Lectura de sensor MPU	45
Tabla 3: Comando para diseño de red GR.....	47
Tabla 4: Comando para diseño de red LVQ	50
Tabla 5: Valores de RECALL y PRECISIÓN para red de señas LVQ	60
Tabla 6: Valores de RECALL y PRECISIÓN para Red Conjugada LVQ.....	63
Tabla 7: Comparación de valores de precisión por letra.	66
Tabla 8: Comparación de valores de recall por letra.	67
Tabla 9. Magnitud F característica	69
Tabla 10: Aplicación de ruido al 0.1% sobre redes neuronales.....	72
Tabla 11: Aplicación de ruido al 1% sobre redes neuronales.....	73
Tabla 12: Aplicación de ruido al 5% sobre redes neuronales.....	74
Tabla 13: Aplicación de ruido al 10% sobre redes neuronales.....	75

RESUMEN

La presente investigación simuló Redes Neuronales Artificiales y evaluó su proceso de clustering de vectores de datos digitales captados por un guante electrónico, pertenecientes a 24 señas estáticas del alfabeto dactilológico peruano, utilizando para este cometido la evaluación de 3 parámetros distintivos como lo son: la tabla de contingencia, la precisión y el recall y se justificó por cuanto posee valor tanto social como académico; así mismo, se explicó la variable independiente, definida por las técnicas de clustering (de Regresión General como las de Aprendizaje por Cuantización Vectorial) a implementar, para ambos casos se optó por trabajar con 3 redes neuronales distintas: una encargada de las señas, otra para los sensores MPU y una red conjugada que congrega los resultados de las redes previas, y la variable dependiente definida por el reconocimiento de los gestos estáticos y evaluada por medio de la tabla de contingencia, la precisión y el recall.

En cuanto a los resultados, evidenciaron que una Red Neuronal de Regresión Generalizada posee una gran capacidad en el reconocimiento de estos gestos, con una precisión y recall del 100% verificada en su tabla de contingencia para datos pre procesados, así como una resistencia al ruido del 1% para mantener esta efectividad; por otro lado, una Red Neuronal de Aprendizaje por Cuantización Vectorial requiere de una mayor cantidad de sensores, así como la adición de una red distinta para poder ser más preciso en su reconocimiento, con valores de precisión y recall iguales a 93.75% y mejorados al 100% con la adición de una red GRNN final. Además de ello, se verificó que esta red neuronal LVQ no posee mucha tolerancia al ruido al tener valores máximos de precisión y recall de 98.14% y 97.43% ante la aplicación de ruido a un nivel del 0.1% y valores mínimos de 90.25 y 88.05 respectivamente ante la aplicación del 10% de ruido.

Palabras clave: Algoritmo de clustering, precisión, recall, Deep Learning Toolbox, Lengua de Señas Peruana

ABSTRACT

The present investigation simulated Artificial Neural Networks and evaluated their process of clustering digital data vectors captured by an electronic glove, belonging to 24 static signs of the Peruvian fingerprint alphabet, using for this purpose the evaluation of 3 distinctive parameters such as: the table of contingency, precision and recall; in addition it was justified because it has both social and academic value, likewise, the independent variable was explained and defined by the Clustering Techniques (General Regression and Learning Vector Quantization) to implement, for both cases, it was decided to work with 3 different neural networks: one in charge of the signs, another for the MPU sensors and a conjugate network that brings together the results of the previous networks; in the other hand the dependent variable was defined by the recognition of static gestures and evaluated through the contingency table, precision and recall.

Regarding of the results, they showed that a Generalized Regression Neural Network has a great capacity in the recognition of these gestures, with a precision and recall of 100% verified in its contingency table for pre-processed data, as well as resistance to noise 1% to maintain this effectiveness; on the other hand, a Learning Vector Quantization Neural Network requires a greater number of sensors, as well as the addition of a different network to be more precise in its recognition, with values of precision and recall equal to 93.75% and improved upon 100% with the addition of a final GRNN network. Furthermore, it was verified that this LVQ neural network does not have much tolerance to noise as it has maximum precision and recall values of 98.14% and 97.43% when applying noise at a level of 0.1% and minimum values of 90.25 and 88.05 respectively before applying 10% noise.

Keywords: Clustering algorithm, precision, recall, Deep Learning Toolbox, Peruvian Sign Language

INTRODUCCIÓN

La presente investigación titulada “Simulación y evaluación de técnicas de clustering para el reconocimiento de gestos estáticos en la traducción de la Lengua de señas peruana” es abordada debido al existente abismo lingüístico entre la comunidad practicante de la Lengua de Señas peruana y la no practicante, y la búsqueda de que nuevas tecnologías, tales como las Redes Neuronales Artificiales en conjunción con “guantes electrónicos”, puedan ser aplicadas en pro de la comunidad. Las principales causas del problema social radican en la continua carencia de políticas inclusivas para la comunidad sorda de nuestro país; la no profesionalización de la lengua oficial reconocida para la comunidad, la Lengua de Señas Peruana y, en consecuencia, una insuficiente cantidad de profesionales capacitados para la interpretación de la Lengua de Señas peruana a lenguaje hablado. Para esta investigación se ha formulado el problema general de la siguiente manera: ¿Es posible simular y evaluar 2 técnicas de Clustering para el reconocimiento de gestos estáticos en la Traducción de la Lengua de Señas Peruana?, interrogante que a través de la investigación se ha dado respuesta.

El interés central de la investigación radica en el estudio del reconocimiento de los gestos estáticos, por parte de las técnicas de clasificación de una Red Neuronal Artificial, para la interpretación de la Lengua de Señas Peruana. Mediante la evaluación de parámetros de cada una de las Redes Neuronales Artificiales implementadas, se pudo identificar características y evaluar el desempeño de cada red neuronal en el trabajo de clasificación de datos característicos de la Lengua de Señas Peruana; en adición a ello existió un interés académico en lo referente a la tecnología de los guantes electrónicos y su aplicación junto con las Redes Neuronales Artificiales, en la interpretación de la Lengua de Señas Peruana.

En el marco de la aplicación de las redes neuronales, la investigación se realizó en línea a lo largo de todas las bases de datos académicas facilitadas por la universidad y otras más, así como también de manuales gratuitos del software utilizado. El estudio expone la correlación entre las técnicas de clustering implementadas y el distintivo reconocimiento y/o clasificación de los datos a los cuales la red neuronal es sometida. El instrumento de recolección de datos fue un “guante electrónico” implementado en un proyecto previo en la Universidad Ricardo Palma y el método radicó en la toma de muestras a 3 personas

distintas realizando las 24 señas estáticas pertenecientes al alfabeto dactilológico peruano; se utilizó el Principio de Pareto para establecer la cantidad de datos necesarios en el entrenamiento y en consecuencia los datos para las pruebas.

Así mismo para un estudio sistematizado del problema, la investigación se ha estructurado de la siguiente manera:

En el Capítulo I se realizó un abordaje al tema de investigación, planteando el problema sobre el cual gira la investigación, la importancia de esta y los objetivos a los que se apunta para resolver la incógnita: ¿Es posible simular y evaluar 2 técnicas de Clustering para el reconocimiento de gestos estáticos en la Traducción de la Lengua de Señas Peruana?

En el Capítulo II brindó la base teórica de cada variable de investigación, además se expuso el desarrollo de las redes neuronales en la ciencia, así como también el tratamiento que ha tenido nuestro país con la Lengua de Señas Peruana en el tiempo; se exponen también los acercamientos entre la ciencia y la Lengua de Señas alrededor del globo y en nuestro país como antecedentes históricos, de manera que se puedan analizar procedimientos y resultados pasados

En el Capítulo III se planteó la manera como se llevó a cabo la experimentación y se detallan las etapas necesarias para el cumplimiento de nuestros objetivos, presentándose también procedimientos, rutinas de programación y datos cuantitativos de la experimentación.

En el Capítulo IV se detalló los resultados finales de la experimentación, junto con la comparación y análisis de los parámetros establecidos para la evaluación de las técnicas de clustering determinadas.

Para finalizar la investigación se redactaron las conclusiones, recomendaciones y se describieron las referencias bibliográficas utilizadas.

CAPÍTULO I: PLANTEAMIENTO Y DELIMITACIÓN DEL PROBLEMA

1.1. Descripción del problema

La pérdida de audición discapacitante es definida por la Organización Mundial de la Salud (OMS) como “... una pérdida de audición superior a 40dB en el oído con mejor audición en los adultos, y superior a 30dB en el oído con mejor audición en los niños.” (OMS, 2019).

En nuestro país, según el Informe Temático N°5 realizado por la Semana Internacional de la Persona con Discapacidad Auditiva “... se podría estimar que la población con discapacidad auditiva al 2015 sería aproximadamente 560 mil 730 personas.” (CONADIS, 2016), lo que equivale al 1.8% de la población total de nuestro territorio y como todo grupo humano, la necesidad de comunicarse es primordial para su desarrollo; en nuestro país el lenguaje reconocido para este grupo es la Lengua de Señas Peruana (LSP).

“La Lengua de Señas Peruana (LSP) es un idioma autóctono de la comunidad Sorda en el Perú. Como todo idioma, tiene un léxico, estructuras lingüísticas propias, modismos, características regionales y otros componentes que caracterizan a un lenguaje viviente.” (Enseñas Perú, 2015).

Es el año 2010 en el que se oficializa la Lengua de Señas Peruana (LSP) “... como lengua de las personas con discapacidad auditiva en todo el territorio nacional.” (El Peruano, 2010) con la Ley N°29535, a la cual se le sumaron posteriormente otras con el fin de fomentar la inclusión de la comunidad; se debe destacar que, en la normativa de la Ley°29535, se contempla la necesidad de intérpretes de LSP a lenguaje hablado; a 5 años de promulgada la Ley, solo existían 23 traductores certificados para la realización de esta labor y otros pocos intérpretes empíricos no profesionalizados (Sausa, 2015). Si bien en 2018 hubo un intento para lograr la profesionalización del LSP, hasta la fecha no es todavía una lengua profesionalizada en nuestro país, por lo que las certificaciones son únicamente internacionales.

En entrevistas con la Srta. Marita Padilla, directora ejecutiva de la ONG Enseñas Perú , se mencionó: que se conocen casos de abusos a personas sordas en distintas comunidades del interior, que no existen intérpretes capacitados en abundancia para la cantidad de casos (la entrevistada es intérprete especializada en este ámbito), teniendo en consecuencia personas sordas sin acceso a la administración de justicia por la carencia de un intérprete para poder brindar sus declaraciones, esto es algo que se refleja también en hospitales, comisarías y colegios, con adultos, jóvenes y niños que padecen sordera. Es menester mencionar que el 1.8% de la población total de nuestro territorio representa a la población con discapacidad auditiva al año 2015 (CONADIS, 2016), este porcentaje de la población según lo especificado por Marita Padilla, no tienen fácil acceso a servicios primordiales como son educación, servicios médicos, entre otros; debido a que el LSP no es muy conocido ni difundido entre la comunidad oyente, así como tampoco se encuentra entre las políticas de inclusión de los gobiernos de turno.

De lo anteriormente expuesto, se puede resumir que el común denominador entorno a la problemática es la brecha lingüística existente entre la comunidad sorda practicante de LSP y la comunidad oyente no practicante de esta lengua; este aislamiento lingüístico se debe a 3 factores importantes: la carencia de políticas inclusivas, si bien en el 2010 se promulgó la Ley N°29535, se tuvieron que esperar unos cuantos años para su reglamentación y aplicación; la no profesionalización del LSP en nuestro país, esto significa que no se puede tener un título profesional para ejercer de intérprete para esta lengua en el territorio nacional; y finalmente la insuficiente cantidad de intérpretes (23) para una población creciente de aproximadamente 560 mil personas, estos pocos intérpretes existentes han sido certificados en el extranjero. Si bien la ley promulgada en el 2010 instaba a las instituciones estatales a contar con intérpretes para la comunidad sorda, esta necesidad no fue abastecida; finalmente la carencia de promoción de la LSP implica ignorancia para la mayoría de personas acerca de esta comunidad y de sus necesidades, hasta en el ámbito político; arriesgando de esta manera a la población sorda, ya que no poseen un fácil acceso a establecimientos de servicios primordiales como lo son hospitales, comisarías, centros educativos, debido a la imperativa necesidad de un intermediario para la comunicación de sus necesidades; junto a ello, muchas personas son separadas de la comunidad al no poder comunicarse, propiciando escenarios de abusos que quedan impunes debido a que no pueden ser comunicados por los violentados.

Frente a esta carencia se vislumbra la posibilidad de aplicar las tecnologías actuales en pro de la comunidad sorda, y la opción más viable en este ámbito, por la capacidad y potencial, es la aplicación de las Redes Neuronales Artificiales (RNA), rama del Machine Learning (“Aprendizaje de maquina” por su traducción al español), que por su capacidad de aprendizaje, pueden desempeñar roles muy cruciales de manera autónoma en el ámbito del reconocimiento de patrones y clasificación (clustering) como muchos autores han comprobado. Existe entonces una relación inminente entorno a las distintas técnicas de clasificación (técnicas de clustering) de las RNA y la capacidad de estas para reconocer y clasificar patrones, para la presente investigación, patrones propios de la Lengua de Señas Peruana (gestos estáticos), para de esta manera poder realizar una apropiada traducción del mismo, lo cual precisa de una evaluación individual de cada una de estas técnicas en base a criterios o medidas de evaluación teóricamente definidas (Walde, 2003).

1.2. Formulación del problema

1.2.1 Problema General

¿Es posible simular y evaluar 2 técnicas de Clustering para el reconocimiento de gestos estáticos en la traducción de la Lengua de Señas Peruana?

1.2.2 Problemas Específicos

- a) ¿Cómo captar y clasificar los datos digitales propios de la LSP mediante una Red Neuronal de Regresión Generalizada (GRNN)?
- b) ¿Cómo captar y clasificar los datos digitales propios de la LSP mediante una Red Neuronal de Aprendizaje por Cuantificación Vectorial (LVQNN)?
- c) ¿Es posible realizar pruebas con los algoritmos implementados para conocer su tabla de contingencias producto de la clasificación de gesticulaciones estáticas propias de la Lengua de Señas Peruana?
- d) ¿Es posible realizar pruebas con los algoritmos implementados para conocer su precisión en la clasificación de gesticulaciones estáticas propias de la Lengua de Señas Peruana?
- e) ¿Es posible realizar pruebas con los algoritmos implementados para conocer su recall en la clasificación de gesticulaciones estáticas propias de la Lengua de Señas Peruana?

1.3. Importancia y Justificación del estudio

En la realidad social de nuestro país existe aún un aislamiento de la comunidad sorda debido al desconocimiento acerca de esta colectividad y sus necesidades, lo que acarrea un gran problema social y los convierte en una población de riesgo; por lo tanto, la importancia de este proyecto radica en evaluar la suficiencia con la que las RNAs son capaces de reconocer gestos específicos de manera que puedan suplir la necesidad de un intermediario para la comunicación de las necesidades de esta comunidad; evitando de esta manera las dificultades existentes como lo son el no tener fácil acceso a servicios primordiales como hospitales, comisarías o centros educativos. Así como también, debido a la limitada bibliografía concerniente a la aplicación de la tecnología de los guantes electrónicos para la solución de este problema, la presente investigación toma una importancia académica.

Es justificable la elección del presente tema debido a un previo acercamiento realizado en el año 2019 con la implementación de los Guantes Electrónicos en el proyecto titulado: “Prototipo de guantes electrónicos para la traducción de la lengua de señas peruana (LSP)”, desarrollado en el Taller de electrónica IV (CE1002) en la Universidad Ricardo Palma; de igual manera, al pertenecer a la comunidad académica de la Universidad Ricardo Palma se cuenta con una licencia académica del software MatLab y con ello acceso a su Toolbox Deep Learning, utilizado en la implementación y entrenamiento de redes neuronales artificiales como una de sus múltiples aplicaciones en el mundo científico, el cual será útil para el desarrollo de las RNAs y la adquisición de datos. En adición a ello, la investigación y consultas bibliográficas se dan en línea, en plataformas abiertas y en su mayoría en la plataforma IEEE Xplore a la cuál se tiene acceso gracias al apoyo del asesor.

1.4. Objetivos de la investigación

1.4.1. Objetivo general

Simular y evaluar 2 técnicas de clustering de manera que puedan reconocer gestos estáticos en la traducción de la Lengua de Señas Peruana.

1.4.2. Objetivos específicos

1. Implementar computacionalmente una técnica de Clustering usando una Red Neuronal de Regresión Generalizada (GRNN) para la clasificación de datos digitales característicos de la Lengua de Señas Peruana (LSP).
2. Implementar computacionalmente una técnica de Clustering usando una Red Neuronal de Aprendizaje por Cuantificación Vectorial (LVQNN) para la clasificación de datos digitales característicos de la Lengua de Señas Peruana (LSP).
3. Evaluar los algoritmos implementados mediante su tabla de contingencia, producto de la clasificación de gesticulaciones estáticas propias de la Lengua de Señas Peruana.
4. Evaluar los algoritmos implementados para conocer su precisión en la clasificación de gesticulaciones estáticas propias de la Lengua de Señas Peruana.
5. Evaluar los algoritmos implementados para conocer su recall en la clasificación de gesticulaciones estáticas propias de la Lengua de Señas Peruana.

1.5. Delimitación del estudio

El presente estudio utiliza un prototipo de guantes electrónicos (en base a sensores Flex y acelerómetros) desarrollado en la Universidad Ricardo Palma durante el año 2019 como instrumento de adquisición de datos, para captar datos digitales propios de las señas de la LSP de manera que al ser usados por dos RNAs previamente entrenadas (GRNN y LVQNN), se pueda hacer una evaluación del proceso de Clustering de ambas redes, analizando la tabla de contingencia, la precisión y el *recall* de cada una de estas, para el caso particular se utiliza un solo guante (mano derecha) de talla mediana para el uso de personas adultas, se debe recalcar que el instrumento no es sujeto de estudio.

Como se mencionó anteriormente, la RNA permite que el usuario no tenga la necesidad de recurrir a un tercero para la interpretación al momento de querer expresarse, enfocaremos este estudio en las señas estáticas del alfabeto dactilológico peruano propias de la LSP de la “Guía para el aprendizaje de la Lengua de Señas Peruana” del Ministerio de Educación del Perú. (DIGEBE, 2015)

CAPÍTULO II: MARCO TEÓRICO

2.1. Marco histórico

El nacimiento de las Redes Neuronales Artificiales (RNA) se debe a la publicación matemática en 1943 de Warren S. McCulloch y Walter Pitts “*A logical calculus of the ideas immanent in nervous activity*”, en la cual fue por primera vez implementada una red neuronal simple con circuitos eléctricos, logrando con ello un ligero acercamiento al funcionamiento de las neuronas en conjunto y el primer modelo de neurona artificial (McCulloch, 1943), de este estudio se concluye también que la respuesta de las neuronas biológicas son del tipo “no lineal”. Para 1949, Donald Olding Hebb expone en 2 capítulos de su libro “*The organization of behavior; a neuropsychological theory*” el comportamiento de las neuronas entorno a su aprendizaje y por tanto define una regla de aprendizaje para las mismas (Hebb, 1949).

En 2003 Walczak nos resume un poco la historia de las RNA en su publicación “*Artificial Neural Networks*”, donde redacta que para 1969 Minsky y Papert, con su libro “*Perceptron*” describen las limitaciones computacionales de las redes neuronales de una sola capa (*single layer*); y para 1986, James Lloyd McClelland y David Everett Rumelhart publican “*Parallel Distributed Processing*”, en el que hablan sobre el algoritmo de aprendizaje de retro propagación (*back propagation learning algorithm*) que permite el entrenamiento de perceptrones multicapa, introduciendo por tanto el concepto de las capas ocultas de las RNA y con ello los perceptrones multi capa. (Walczak, 2003).

Posteriormente y debido a la historia de su desarrollo, surgió el concepto de Red Neuronal poco profunda (*shallow neural network*). La red neuronal inició como una red mono capa, evolucionó a una red neuronal poco profunda y finalmente a la red neuronal profunda (es necesario comentar que a la red multicapa también se le llama red neuronal profunda) (Kim, 2017).

La Lengua de Señas Peruana (LSP) es definida según la Ley N°29535 como “... lengua de las personas con discapacidad auditiva en todo el territorio nacional.” (El Peruano, 2010), otro concepto de este, pero desde la parte técnica lo define como “Un lenguaje que se percibe a través de la vista y requiere el uso de las manos como articuladores activos,

el uso del espacio como lugar de articulación (estructura fonológica) y como referencia temporal.” (DIGEBE, 2015).

Es promulgada en el año 2010 la Ley N°29535 con la cual es reconocida legalmente la LSP a nivel nacional, actualmente en nuestro país, el aislamiento de la comunidad sorda es todavía un hecho de suma importancia, y esta situación viene dada por factores como el desconocimiento colectivo acerca de esta comunidad y sus necesidades, ligado también a la falta de políticas inclusivas para este porcentaje poblacional, lo que deviene en una grave problemática social. Se tienen registros de publicaciones del Ministerio de Educación de los años 1987 y 1996 concernientes al tema de la difusión de la LSP. (Fundación Telefonica, 2017) (Otazú, 26).

2.2. Investigaciones relacionadas con el tema

Se pudo tener acceso a algunos estudios previos concernientes al tema, algunas de las soluciones planteadas ante esta problemática serán expuestas en la siguiente sección.

El Departamento de Ingeniería de la Universidad de LaSalle, presentó en 2014 el artículo “Glove-Based Sign Language Recognition Soution to Assist Communication for Deaf Users”, del autor López-Noriega, donde utiliza un perceptrón muticapa de 5 entradas y 26 salidas, una para cada letra del abecedario, y compara una cantidad de redes neuronales. Según los resultados obtenidos, las redes neuronales entrenadas demostraron un buen funcionamiento y reconocieron las señas de manera óptima; sin embargo, el algoritmo de propagación rápida demostró un mejor desempeño y consideran implementarla para un futuro prototipo final. (López-Noriega, 2014)

Para el mismo año, en la Universidad de Cornell, se presentó un proyecto similar, “Sign Language Glove”, en el que se implementa un guante con sensores flex y acelerómetro, con los que se captan señales del Lenguaje de Señas Americano (ASL), y que junto con un algoritmo de Machine Learning como el Support Vector Machine (SVM), identificaron 28 grupos de gestos y lograron un proceso de clustering exitoso al 98% debido a que necesitaban mejoras para evitar la ambigüedad entre las señas de las letras “u” y “v” (Lin Monica, 2014).

Otra de las opciones exploradas fue un sistema de reconocimiento de lenguaje de señas usando sensores 3D en “A real-Time ASL Recognition System using Leap Motion Sensors”, del departamento de Ingeniería Electrónica e Informática de la Universidad Politécnica de Hong Kong, donde usando lo que denominan MSDF por las siglas en inglés de Multiple Sensors Data Fusion; el procesamiento de los datos se lleva a cabo por medio de Modelos ocultos de Markov (HMM). Los resultados presentados demuestran una alta tasa de reconocimiento para lo propuesto en comparación con los sistemas de un solo sensor. (Kai-Yin, 2015).

Una contribución nacional proviene de la ciudad de Lambayeque con una tesis para la obtención del título profesional de Ingeniero Electrónico de la Facultad de Ciencias Físicas y matemáticas de la Universidad Pedro Ruiz Gallo, la tesis es titulada “Diseño y construcción de un prototipo de sistema electrónico para conversión de lenguaje de señas a mensajes de voz para la comunicación de personas sordomudas, en la ciudad de Chiclayo”. En la tesis se implementan guantes con sensores Foto-Flex, los cuales van a tener la función de recabar información de los dedos flexionados de cada seña realizada, junto con acelerómetros; la RNA utilizada en esta investigación es una red de Retropropagación (Back Propagation, BP). La Red fue evaluada entorno a su precisión, obteniendo una clasificación satisfactoria del 97%. (Fernández Suárez, 2017).

Otra de las soluciones mayormente exploradas para la problemática en el mundo científico es planteada con el procesamiento digital de imágenes, para el cual distintos autores proponen las siguientes opciones:

Csoka propone su artículo “Key Frame xtraction from video sequences Containing ASL Signs with concealed Transmission Errors” del Instituto de Información Multimedia y Tecnologías de comunicación de la Facultad de Ingeniería Eléctrica y Tecnologías de Información, donde se realiza la extracción de datos de secuencias de videos que contienen signos propios del ASL (American Signal Language) por medio de un algoritmo capaz de detectar y seguir el trayecto de una mano, identificando por cada “frame” si existe un signo deseable e importante, los resultados se enfocan más en la capacidad de seguir la mano correctamente, lo cual se ve afectado por la calidad del video

y/o distorsiones del mismo. La red es evaluada conforme su precisión, recall y magnitud-F1, con los que sustentan numéricamente sus conclusiones. (Csoka, 2017).

Otra propuesta para secuencias de videos es expuesta en “Dynamic Tool for American Sign Language Finger Spelling Interpreter” por Prateek, del Departamento de Ciencias de la información e Ingeniería de la Universidad de Ingenierías y ciencias B.V. Bhoomaraddi, esta vez se usan secuencias que contienen letras del alfabeto en el Lenguaje de Señas Americano ASL, aquí se utiliza una Red Neuronal Convolutiva (CNN) la cuál se encarga de construir modelos de clasificación para cada una de las 26 letras del vocabulario. Los resultados muestran que la tasa de efectividad de este modelo es del 98.66%, pero se debe tener en cuenta que solo reconoce las letras del alfabeto y puede ayudar para cuando las personas deletrean determinadas palabras. (Praatek, 2018).

Para imágenes estáticas, tenemos 2 referentes, como lo son Keerthi del Departamento de Ingeniería Biomédica de la Universidad SRM en Chennai, India, con el artículo “Software based sign language converter”, el cual utiliza el software LabVIEW para su cometido. La entrada del programa son los gestos de la mano captada por una cámara y posteriormente la salida es en forma de letras en la misma pantalla del computador. El programa puede funcionar muy bien para la identificación de números gesticulados del 1 al 10. (Keerthi, 2016).

Finalmente, Thalange, del Departamento de Ingeniería Electrónica y Telecomunicaciones del Instituto de Tecnología Walchand propone también, en “Effect of thinning extent on ASL number recognition using open-finger distance feature measurement technique” un sistema de reconocimiento de números del 0 al 10 en ASL y se utiliza un algoritmo de adelgazamiento para el reconocimiento de los gestos realizados, así como una red neuronal para la clasificación e identificación de estos. De este se puede concluir, en palabras de los autores, que “el método funcionó perfectamente para números del 0 al 5, adhiriendo mejoras, el rango de reconocimiento para los números del 6 al 9 puede ser incrementado”, así como también que el factor de adelgazamiento es indispensable para la tasa de acierto, las conclusiones son sustentadas por medio de un promedio del porcentaje de precisión en la clasificación de cada categoría. (Thalange, 2015).

Del mismo autor extraemos algunos datos conceptuales, como lo es que, los gestos de este lenguaje pueden dividirse o agruparse en dos categorías y estas, por sus características principales, pueden ser analizadas por métodos definidos, como lo son:

- Gestos dinámicos: Hidden Markov Models (HMM), Dynamic Time Warping, Bayesian Networks u otro método de reconocimiento de patrones.
- Gestos estáticos: Template matching, Eigen spaces o Principal Component Analysis, Elastic Graph Matching, Neural Network u otra técnica de reconocimiento de patrones estándar.

2.3. Base teórica

2.3.1. Técnicas de Clustering

“Técnicas de agrupación que consideran las tuplas de datos como objetos. Dividen los objetos en grupos o clusters, de manera que los objetos dentro de un mismo grupo son similares entre sí y distintos a otros pertenecientes a otros grupos. ...La efectividad de las técnicas dependen de la naturaleza de los datos. Es más eficaz para información que puede ser clasificada en distintos grupos que para datos difusos. Existen muchas medidas para definir un cluster y la calidad de este.” (Han, 2012).

2.3.1.1. Red Neuronal de Regresión Generalizada: GRNN

Modelada por Donald F. Specht en 1991, en su artículo “*A General Regression Network*”, da el alcance de una “red la cual usa un método que la libera de la necesidad de asumir una forma funcional específica. De manera que, le concede la forma apropiada para ser expresada como una función de densidad probabilística (*probability density function*, pdf) la cual es determinada empíricamente de los datos observados utilizando una estimación de ventana Parzen... La ecuación de regresión resultante puede ser implementada en paralelo con una estructura semejante a la de una red neuronal. Los parámetros de la estructura son determinados directamente de los ejemplos a los que es sometidos más que de las iteraciones, la estructura “aprende” y puede empezar a generalizar de manera inmediata.” (Specht, 1991). En el mismo artículo, se describe la arquitectura multicapa de la red, las distintas capas o unidades neuronales (en términos del autor) con las que cuenta son: unidad de entrada, encargada de la distribución, provee a las

neuronas contiguas de valores numéricos; unidad de patrones, dedicada a un ejemplar o un centro de grupo específico, al ingresar un vector X , se resta del vector almacenado que representa a cada centro de grupo; unidad de sumatoria, esta realiza la operación producto escalar entre el vector de pesos y el vector resultante de la unidad de patrones; y finalmente la unidad de salida.

El software Matlab diagrama esta red según lo mostrado en la Figura N° 1, en donde se observa la “Capa de entrada”, la “Capa de Base Radial” (aquí es donde se encuentra la red GR) y finalmente una “Capa Lineal Especial”. Aquí, se denota con R al número de elementos del vector de entrada, S^1 al número de neuronas competitivas y S^2 el número de neuronas lineales.

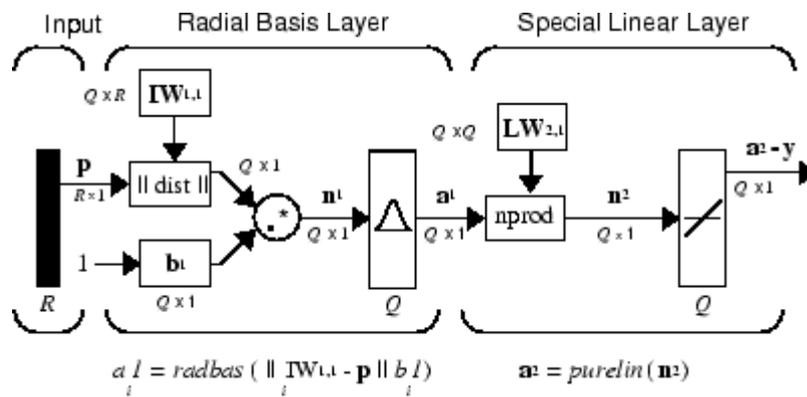


Figura N° 1: Arquitectura de Red Neuronal GR

Fuente: (MathWorks, 2020)

2.3.1.1.1. Redes multicapa o *Multi Layer*

Cuando son adicionadas capas ocultas a una red monocapa, a la red neuronal que posee una sola capa oculta se llama red neuronal poco profunda o superficial (*shallow neural network*) y las que poseen más de una capa oculta son llamadas redes neuronales profundas (*deep neural network*) (Kim, 2017). En palabras de otro autor, se manifiesta que: “estas redes se distinguen a sí mismas por la presencia de una o más capas ocultas cuyos nodos computacionales son llamados neuronas ocultas; el término “oculto” refiere al hecho de que esta parte de la red neuronal no es vista directamente por la entrada o salida de la red” (Haykin, 2008).

A continuación, se presenta en la Figura N°2 una red multicapa, precisando, la graficada es una “red neuronal poco profunda” ya que cuenta con una sola capa de neuronas ocultas.

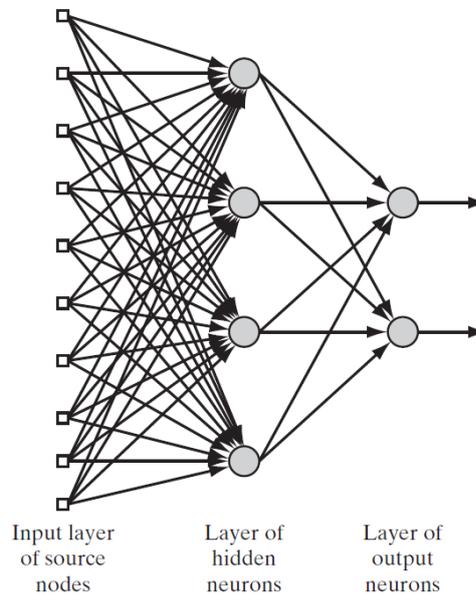


Figura N° 2: Estructura de capas de una red Multicapa.

Fuente: (Haykin, 2008)

2.3.1.1.2.Capa de Base Radial

En la Figura N° 1, en la capa de base radial, el bloque $\| \text{ndist} \|$ tiene como entradas al vector “p” y la matriz de pesos iniciales “IW”, dando como resultado el vector con S^1 elementos. El vector bias b^1 y la salida del bloque $\| \text{ndist} \|$ son combinados con la operación “.*” del software MatLab, la cual realiza una multiplicación de “elemento por elemento” emulando de esta manera el producto escalar realizado en la unidad de sumatoria.

2.3.1.1.3.Capa Lineal Especial

En esta capa, el bloque “nprod” mostrado en la Figura N° 1, produce los elementos S_2 pertenecientes al vector n_2 . Cada uno de estos elementos es el producto escalar de una fila de los pesos “LW” (pesos pertenecientes a la segunda capa) con el vector de entrada a_1 . Desempeñando posteriormente una función similar a la capa lineal de la red LVQ.

2.3.1.2. Red Neuronal de Aprendizaje por Cuantificación Vectorial: LVQNN

En términos de Bonifacio del Brio, la Red LVQ, es un modelo supervisado para llevar a cabo tareas de clasificación de patrones. Esta se compone de una capa simple de neuronas Kohonen (arquitectura de red monocapa); supondremos de entrada que cada una de las “n” neuronas representa una clase, y almacena un vector de referencia “ w_j ” que representa el prototipo de clase. Cuando se presenta a la red un patrón de entrada “x”, compuesto por “m” entradas, cada neurona calcula la distancia que separa su vector de referencia del de entrada, la neurona “c” de menor distancia con el vector de referencia “ w_c ”, indica la clase “C” en la que el patrón “x” queda clasificado. (del Brio, 2002).

Es idóneo y pertinente hacer una distinción de los conceptos de Cuantización Vectorial (*Vector Quantization*, VQ), Mapas Auto Organizados (*Self-Organized Maps*, SOM) y Red Neuronal de Aprendizaje por Cuantificación Vectorial (LVQ), esta distinción la hace Kohonen en su libro de Mapas Auto Organizados y una publicación del mismo nombre, en la que postula: “los mapas auto-organizados (SOM) son usados para clasificación de patrones en los cuales, las celdas y respuestas son agrupados, cada uno correspondiendo a una clase discreta de patrones, el problema se convierte en un proceso de decisión y tiene un tratamiento particular. El mapa original, como cualquier método de cuantización vectorial (VQ), intenta aproximar el valor de la señal de entrada, o su función de densidad probabilística por medio de vectores “marcadores” cuantizados, localizados en el espacio cercano a la entrada para minimizar el error. Si las señales a ser clasificadas tienen un número finito de categorías, varios de estos vectores marcadores son generados para representar cada grupo y su pertenencia a dentro de cada grupo ya no es importante. De hecho, solo las decisiones hechas en los bordes (de cada grupo) toman importancia... Esta estrategia y algoritmo de aprendizaje es introducido por el presente autor y llamado LVQ.” (Kohonen, 1990).

La figura N°3 muestra la arquitectura de una Red Neuronal LVQ planteada en el software Matlab, se puede visualizar la capa de entrada, una Capa competitiva (la base de la red LVQ) y finalmente una “Capa Lineal”. Ambas capas, son conformadas por una neurona por cada clase u objetivo respectivamente (en otras palabras, poseen una arquitectura monocapa). Aquí, se denota con R al número de

elementos del vector de entrada, S1 al número de neuronas competitivas y S2 el número de neuronas lineales.

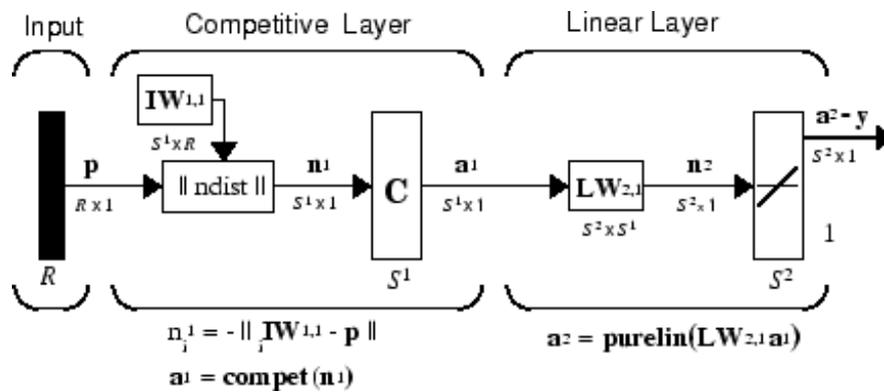


Figura N° 3: Arquitectura de Red Neuronal LVQ

Fuente: (MathWorks, 2020)

2.3.1.2.1.Red Monocapa o Single Layer

Red neuronal simple con únicamente una capa de entrada y una capa de salida (sin capas ocultas). (Kim, 2017) tal y como se puede observar en la Figura N°4, en la cual se presentan las “Capas de entrada” y las “Capas de salida de las neuronas”.

Para Haykin, la designación “monocapa” refiere a la capa de salida de los nodos computacionales (neuronas). No considera la capa de entrada debido a que no existe labor computacional en dicha capa. (Haykin, 2008).

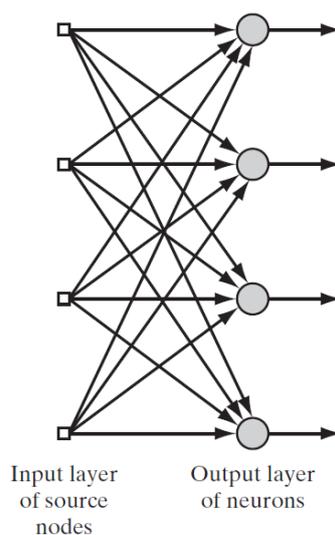


Figura N° 4: Estructura de capas de una red Monocapa.

Fuente: (Haykin, 2008)

2.3.1.2.2.Capa Competitiva

Redes en las que solamente una neurona (o grupo de vecinas) puede quedar finalmente activadas. La base de operación de estos modelos es la competición entre neuronas, materializada en forma de inhibiciones laterales, a través de las cuales cada una trata de inhibir a las demás. En este proceso de competición la neurona más activada conseguirá inhibir a todas las demás, por lo que será la única que permanezca activada, motivo por el cual estas redes también se denominan redes WTA (*winner take all*). (del Brio, 2002).

En la Figura N° 3, en la capa competitiva, se observa que el bloque $\| \text{ndist} \|$ procesa el vector de entrada “p” y la matriz de pesos iniciales “IW”, dando como resultado el vector con S^1 elementos. En el caso particular, la regla de aprendizaje utilizada es LVQ1, en la cual, al ser un vector “p” presentado junto con la matriz de pesos iniciales, se obtiene un vector de distancia; con esto las neuronas ocultas comienzan a competir. Suponiendo que un elemento “ i_{th} ” tiene el resultado “más positivo”, obteniendo como ganadora de la competencia a la neurona i^* , la función de transferencia competitiva produce un 1 como el i^*_{th} elemento de a^1 (salida de la capa competitiva).

2.3.1.2.3.Capa Lineal

Se encarga de transformar las clases o grupos de la capa competitiva en objetivos de clasificación definidos por el usuario. Al aprender las neuronas de la capa competitiva las subclases del espacio de entrada que pertenecen a la clase objetivo de la capa lineal adquieren unos pesos LW, con valor 1 para neuronas n^2 en la capa lineal y 0 para el resto de las neuronas de la misma capa. Es de esta manera que la capa lineal produce un 1 al ganar alguna neurona la competencia, otorgando una salida única y convirtiendo las subclases de la capa competitiva en clases objetivo. En resumen, un 1 en una fila i_{th} de la salida de la capa competitiva a_1 podrá recoger de manera eficiente una columna de LW como la salida de la red, correspondiendo estos 1 a una clase específica.

2.3.2. Reconocimiento de Gestos Estáticos

2.3.2.1. Evaluación de algoritmos

Medidas independientes y confiables para la evaluación y comparación de experimentos y resultados de agrupamiento. (Walde, 2003)

Un artículo se explaya más en el tema y menciona 4 propiedades indispensables para mediciones deseables sobre el rendimiento de sistemas de recuperación de información, es así como el autor (Swets, 1969) numera estas propiedades de la siguiente manera:

- Primero, debe únicamente expresar la habilidad del sistema para distinguir entre objetos deseados e indeseados, la medida debe ser solo de eficacia, dejando para una consideración separada, factores como la eficiencia.
- Segundo, no debe ser aturdida por la relativa disposición del sistema para emitir objetos, esto es que debe expresar el poder de discriminación independientemente de cualquier criterio de aceptación empleado (ya sea este una característica propia del sistema o sea un valor ajustado por el usuario).
- Tercero, la medida debe ser un único número, de manera que pueda ser transmitida de manera simple e inmediatamente perceptible.
- Cuarto, la medida debe permitir el ordenamiento de distintas ejecuciones, indicaría en cantidad la diferencia que separa dos ejecuciones distintas, y evaluaría la actuación de cualquier sistema en términos absolutos, esto es, que la métrica tendría una escala de una unidad, un cero verdadero y un valor máximo.

El mismo autor concluye aclarándonos que “cumpliendo las propiedades, se puede estar seguro de contar con un indicador puro y válido sobre que tan bien un sistema (o método) está realizando la función o tarea para la que fue diseñado.” (Swets, 1969). Y denota la “tabla de contingencia”, “precisión” y “recall” como principales parámetros para el análisis de las técnicas de Clustering, y son cuantiosamente usados en distintas evaluaciones sobre estos sistemas de categorización (como ejemplo, se pueden tomar algunos artículos referenciados por Walde en su artículo del 2003).

2.3.2.1.1. Tablas de contingencia

En el artículo “*Effectiveness of information Retrieval Methods*” del autor John A. Swets, se determina que: “La información presentada por una medida de teorías de decisión, son entradas en una tabla de contingencia 2x2; de modo que los hechos que, correcta o incorrectamente, respondan a las preguntas realizadas podrán ser presentados o retenidos”. (Swets, 1969)

Del mismo modo, Walde nos dice que las tablas de contingencia son un medio típico para describir y definir la asociación entre particiones. La definición formal que el mismo autor nos presenta en su publicación es:

“Definición 4.2 Una tabla de contingencias $C \times M$ es una matriz de dimensión $C \times M$ con filas C_i , $1 \leq i \leq k$ y columnas M_j , $q \leq j \leq l$. los elementos t_{ij} denota el número de objetos que pertenecen al conjunto C_i en la partición C (resultado del clustering) y al conjunto M_j en la partición M (la clasificación manual). La suma de los valores de filas o columnas nos da los valores marginales $t_{i.}$ y $t_{.j}$, los que refieren al número de objetos en las clases C_i y M_j , respectivamente. La suma de los valores marginales resulta en el umero total de objetos “ n ” usados en el proceso de clustering.” (Walde, 2003)

Se grafica el modelo de tabla de contingencia definido anteriormente en la Figura N°5.

	M_1	M_2	
C_1	t_{11}	t_{12}	$t_{1.}$
C_2	t_{21}	t_{22}	$t_{2.}$
C_3	t_{31}	t_{32}	$t_{3.}$
	$t_{.1}$	$t_{.2}$	n

Figura N° 5: Modelo de tabla de contingencia $C \times M$

Fuente: (Walde, 2003)

2.3.2.1.2. Precisión

“Proporción de elementos recuperados que se consideran relevantes, y también puede ser considerado como “la estimación de la probabilidad condicional de que un elemento relevante sea recuperado.” (Swets, 1969)

Un poco más extenso, Walde lo considera como “la equivalencia de clases en el clustering y calcula la existencia y carencia de objetos entre la clasificación manual con respecto a la del clustering.” (Walde, 2003)

La expresión matemática (1) describe la ecuación definida por el autor para hallar el valor numérico de la precisión:

$$precisión = \frac{t_p}{f_p + t_p} \quad (1)$$

En la cual:

tp = verdaderos positivo

fp = falsos positivos

2.3.2.1.3.Recall

“Proporción de ítems relevantes recuperados, y puede tomarse también como una estimación de la probabilidad condicional de que sea recuperado un elemento relevante.” (Swets, 1969)

Walde define este parámetro como “la proporción de pertenencia de objetos existentes en un grupo relevante comparado con el mínimo número de objetos correctos pertenecientes.” (Walde, 2003)

Para hallar el recall de una red neuronal, el autor define la ecuación descrita en la expresión matemática (2):

$$recall = \frac{t_p}{f_n + t_p} \quad (2)$$

En la cual:

tp = verdaderos positivo

fn = falsos negativos

2.3.2.1.4.Magnitud-F

Matemáticamente modelado por el autor Rijsberger en 1979 con su obra “Information Retrieval” no es la ecuación presentada como tal hasta 1992 con la autora Nancy Chinchor en su publicación “MUC-4 evaluation metrics” donde menciona que la Magnitud-F “proporciona una forma de combinar el recall y la precisión para obtener una única medida que se encuentre entre estos dos parámetros.” (Chinchor, 1992)

La autora define esta relación según la ecuación descrita en la expresión matemática (3):

$$\text{Magnitud} - F = \frac{(\beta^2 + 1) * PR}{\beta^2 P + R} \quad (3)$$

Donde:

P= precisión

R= recall

β = importancia entre recall y precisión

2.3.2.2. Gestos estáticos

Gesticulaciones propias del Lenguaje de señas, las cuales no implican variación en el tiempo o movimiento. Entre las señas propias del alfabeto dactilológico peruano se pueden contabilizar 24 señas estáticas (se puede intuir que las 3 señas restantes implican movimientos, siendo por tanto gestos dinámicos, estas corresponden a las letras J, Ñ y Z). Las 27 señas son graficadas en su totalidad en la Figura N°6.



Figura N° 6: Alfabeto dactilológico peruano.

Fuente: (DIGEBE, 2015)

2.4. Definición de términos básicos

2.4.1. Arquitecturas de Redes Neuronales

Se denomina arquitectura a la topología, estructura o patrón de conexionado de una red neuronal. En un sistema neuronal artificial o ANS (por sus siglas en inglés) los nodos se conectan por medio de sinapsis, esta estructura de conexiones sinápticas determina el comportamiento de la red. (del Brio, 2002)

2.4.2. Clustering o Agrupamiento:

Modelo neuronal no supervisado en el cual la red se compone de neuronas de salida discreta, donde cada una representa una categoría y solamente una de ellas puede permanecer activa a la vez. La neurona activa indica a que categoría o grupo (*cluster*) pertenece, a partir de la medida de distancias entre patrones presentados. (del Brio, 2002)

2.4.3. Clusters o Grupos

Asociación de neuronas de una capa, también denominado grupo neuronal, dentro de estos las neuronas suelen ser del mismo tipo. (del Brio, 2002)

2.4.4. MATLAB: Deep Learning Toolbox

Marco para diseñar e implementar redes neuronales profundas con algoritmos, modelos previamente entrenados y apps. (MathWorks, 2020).

2.4.5. Neurona Artificial

Es definida formalmente como un “dispositivo simple de cálculo que, a partir de un vector de entrada procedente del exterior o de otras neuronas proporciona una única respuesta o salida.” (del Brio, 2002), por otro lado, las neuronas son también definidas en síntesis como “sistemas altamente paralelos no lineales y densamente interconectados y basados en procesadores simples: modelo simplificado de la neurona biológica.” (Huamaní, 2008).

2.5. Diseño de la Investigación

2.5.1. Variables de Investigación

2.5.1.1. Independiente

Técnicas de Clustering: “...técnicas de agrupación que consideran las tuplas de datos como objetos. Dividen los objetos en grupos o clusters, de manera que los objetos dentro de un mismo grupo son similares entre sí y distintos a otros pertenecientes a otros grupos” (Han, 2012).

2.5.1.2. Dependiente

Reconocimiento de gestos estáticos: Óptima clasificación de gestos propios de la LSP que no presentan movimiento de las manos o variación en el tiempo.

2.5.2. Tipo y Método de investigación

La presente investigación será un estudio correlacional entre las técnicas de clustering y el proceso de reconocimiento y/o clasificación de los datos presentados; con un enfoque cualitativo, puesto que los datos que serán recolectados son observables y medibles.

Hernández Sampieri define entre los diseños de investigación a los experimentales y no experimentales, los cuales son definidos respectivamente como “estudios en los que se manipulan intencionalmente una o más variables independientes, para analizar las consecuencias que la manipulación tiene sobre una o más variables dependientes” y “investigación en la que se observan fenómenos tal y como se dan en su contexto natural, para después analizarlos.” (Hernandez, 2014), en consecuencia, afirmamos que la presente investigación tiene un diseño experimental debido a la manipulación de su variable independiente y posterior observación de su influencia sobre la dependiente.

2.5.3. Técnicas e instrumentos de recolección de datos

El instrumento utilizado en la recolección de datos fue un “guante electrónico” diseñado en la Universidad Ricardo Palma durante el ciclo académico 2019-II en el proyecto titulado “Prototipo de guantes electrónicos para la traducción de la lengua de señas peruana (LSP)”, el cual está conformado por sensores “Flex”, los cuales tienen la función de representar a cada dedo de la mano, y un sensor MPU6050, del cual nos servimos para poder conocer la orientación espacial de la mano.

2.5.4. Procedimiento para la recolección de datos

Para este punto se debe asegurar que los sensores flex estén correctamente dispuestos a lo largo de cada dedo de la mano cubriendo las falanges proximales, medias y parte de las distales de manera que puedan representar de manera adecuada la flexión de cada dedo (se debe tener especial cuidado en los dedos meñique y pulgar).

Se tomaron a 3 personas para la realización de cada seña prevista, de estas se tomaron 200 muestras, haciendo un total de 600 muestras para cada una de las 24 señas estáticas del alfabeto dactilológico peruano; utilizando el Principio de Pareto y de manera aleatoria, de estas 600 muestras, 480 (80%) fueron destinadas al entrenamiento de las redes neuronales y las 120 (20%) restantes fueron utilizadas en las pruebas.

Finalmente, la lectura digital de los datos proporcionados por los sensores de los guantes electrónicos se realizó mediante microcontrolador “Arduino nano”, el cual también envió dichos datos al software Matlab vía USB, dentro del mismo software se realizó un filtrado de los datos de entrada y posteriormente el entrenamiento y pruebas de las Redes neuronales.

CAPÍTULO III: DISEÑO DE INGENIERÍA

3.1. Diagrama de bloques general

Teniendo en consideración (y como fue anteriormente mencionado) que el guante electrónico es un proyecto previo desarrollado en el año 2019, por un grupo de trabajo al cual perteneció el autor, titulado: “Prototipo de guantes electrónicos para la traducción de la lengua de señas peruana (LSP)” para el curso de Taller de Electrónica IV (CE1002), en el ciclo 2019-II, en la Universidad Ricardo Palma; estos guantes electrónicos implementados constan de sensores flex ubicados en los dedos, emulando las falanges de las manos para la captación del movimiento de flexión de los dedos, un módulo MPU6050 que capta la afección de la gravedad sobre este sensor y un Arduino NANO que se encarga de la lectura de los sensores anteriormente mencionados; en este proyecto previo, la programación y procesamiento de la señal se llevó a cabo en un computador utilizando el software LabVIEW. Se muestra en la Figura N° 7 uno de los guantes electrónicos implementados.



Figura N° 7: Prototipo de guantes electrónicos para la traducción de la LSP

Fuente: Elaboración propia

Se muestra a continuación en la Figura N° 8 el diagrama de bloques general del proyecto, en donde se puede apreciar a grandes rasgos el proceso mediante el cual se lleva a cabo el reconocimiento de los gestos estáticos.

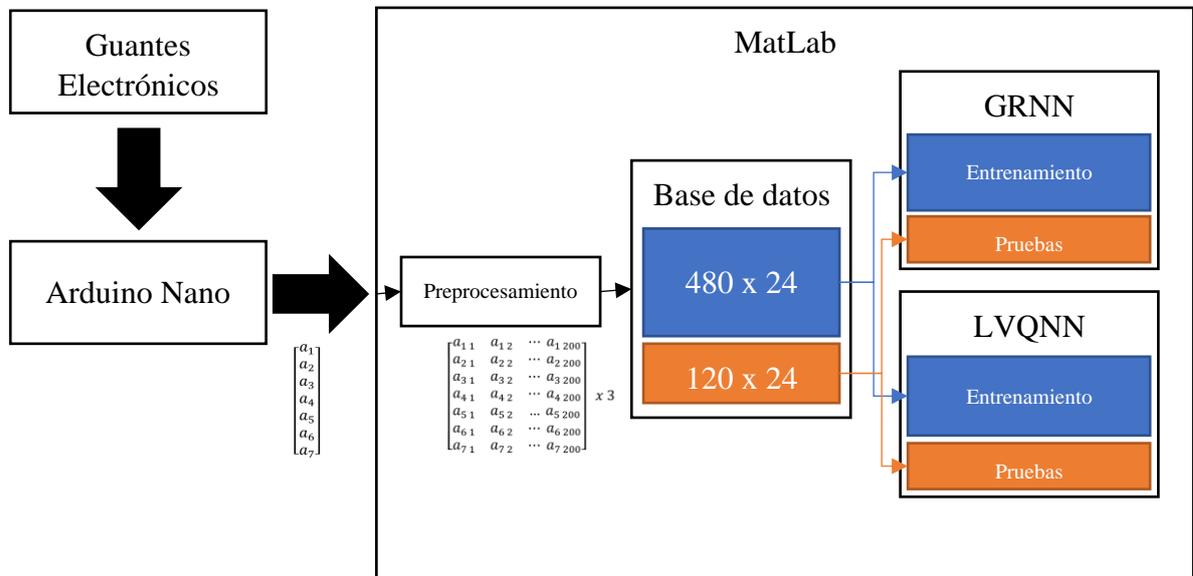


Figura N° 8: Diagrama de bloques

Fuente: Elaboración propia

En primera estancia y como se puede apreciar en la Figura N°8, el Arduino Nano recoge los valores obtenidos de los sensores Flex y MPU (a_1, a_2, \dots, a_7) y los “escribe” en su puerto serial, a continuación, y mediante un programa en el software MatLab se realiza una lectura del puerto serial al cual se encuentra conectado el microcontrolador y se ordenan los datos en matrices de 7x1. Una vez recopilados los datos del puerto serial, se realiza un preprocesamiento antes de almacenarlos en matrices de 7x200 (por cada uno de los 3 individuos) que posteriormente se unirán en una matriz final de 7x600 que definirá a una señal en concreto, formando de esta manera una base de datos.

Una vez creada la base de datos, se seleccionaron de manera aleatoria el 80% de muestras de cada señal para el entrenamiento de ambas redes neuronales, procedimiento que debe anteceder necesariamente a las pruebas; el 20% restante fue utilizado para las pruebas finales de cada una de las redes neuronales para los resultados finales.

3.2. Diagrama de Flujo de la programación en software

Se muestra a continuación en la Figura N° 9 el diagrama de flujo del proyecto que describe el proceso lógico con el cual se lleva a cabo el reconocimiento de los gestos estáticos mediante el software MatLab.

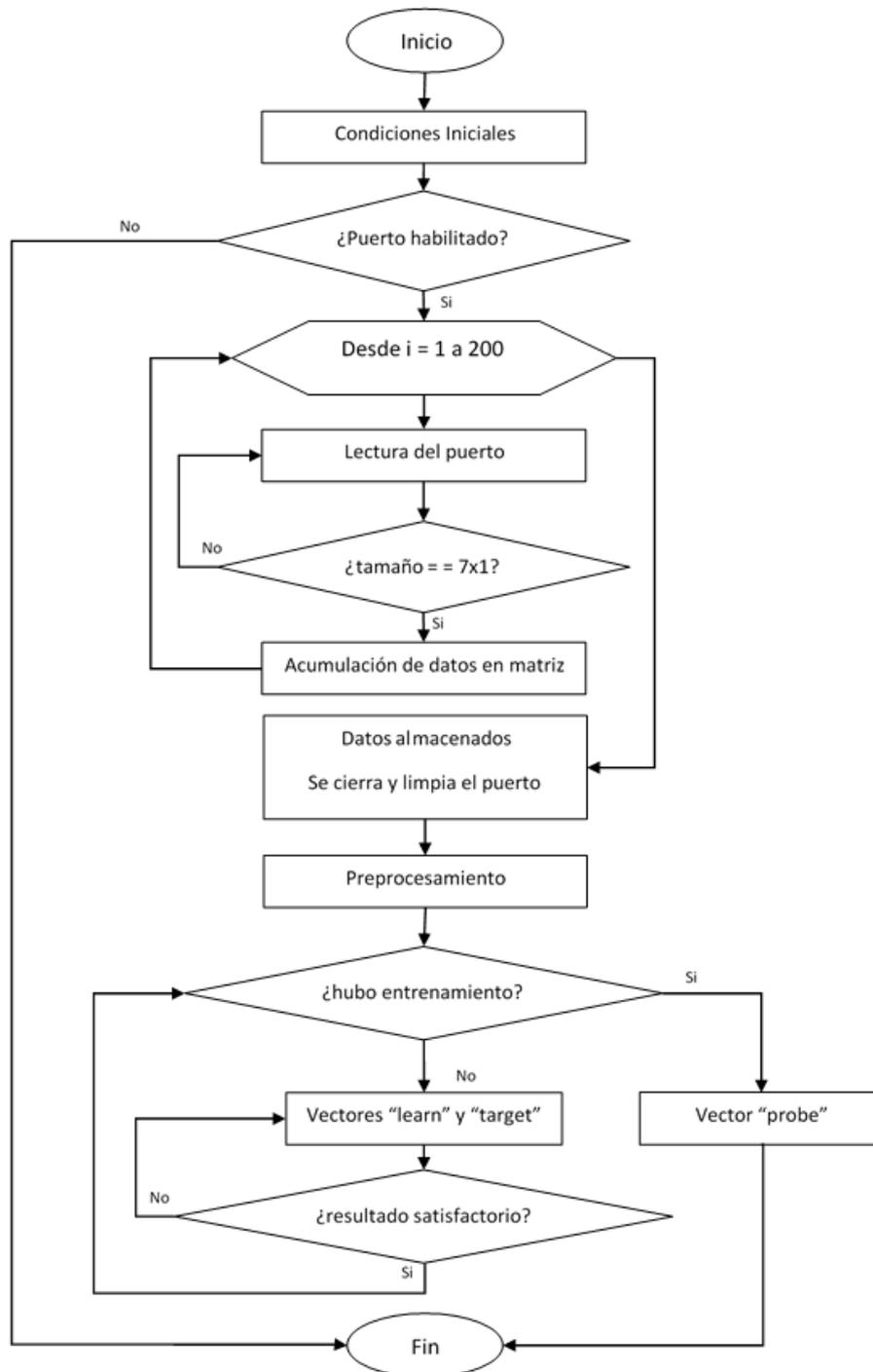


Figura N° 9: Diagrama de flujo

Fuente: Elaboración propia

En una primera sección se describe el programa mediante el cual MatLab “lee” los datos “escritos” en el puerto serial por el microcontrolador, en donde inicia evaluando la habilitación del puerto, es decir, si está conectado el microcontrolador, de estar conectado, se inicia la toma de las primeras 200 muestras para el individuo 1, una a una y verificando que los datos recibidos lleguen de 7 en 7; acabado este proceso se almacena los datos en un archivo “.mat”; se repite el mismo proceso para los individuos 2 y 3.

La siguiente sección es la del preprocesamiento, en la cual se haya una media y se eliminan los errores en la toma de datos, como sobre picos producto de movimientos; al mismo tiempo, es en esta sección en la cual se juntan las matrices de los 3 individuos en una sola, denominada con la letra que caracterizan en mayúsculas (es decir, “AA” para las señas de la letra “A”, “RR” para la letra “R”).

En la última sección, se realiza la separación aleatoria de muestras, como se mencionó anteriormente, se sigue el Principio de Pareto para el cometido, separando entonces el 80% de muestras que conformaran el vector “learn”, el vector “target” es definido manualmente y le indica a la red neuronal las clases u objetivos en los cuales la red debe clasificar a los distintos conjuntos de datos otorgados, para con ello definir los pesos pertinentes para cada neurona. Si al final del entrenamiento no se tiene un error lo suficientemente bajo, se vuelve a entrenar la red, de manera que el error disminuya; caso contrario, se culmina con la fase de entrenamiento y se utiliza el 20% restante del total de muestras para conformar el vector “probe”, el cual es utilizado para las pruebas de las redes neuronales, de los resultados de esta prueba se obtendrán los índices requeridos para el estudio del reconocimiento de los gestos estáticos de cada una de las redes neuronales.

3.3. Acondicionamiento de datos e implementación computacional

En la presente sección se describe de manera detallada los procesos descritos anteriormente para los fines del estudio, para lo cual se emplearon 3 divisiones concretas, siendo la primera la que abarca el proceso de obtención de muestras y el acondicionamiento de estas, la segunda parte desarrolló lo referente a la Red Neuronal GR y finalmente, en la tercera parte, se desarrolló lo pertinente para la Red Neuronal LVQ.

3.3.1. Obtención y acondicionamiento de muestras de entrada

Para una correcta obtención de datos y como se mencionó anteriormente, se deben colocar los sensores flex a lo largo de cada dedo de la mano cubriendo las falanges proximales, medias y parte de las distales tal y como se muestra en la Figura N° 10, esto se hace debido a que se necesita tener un valor numérico preciso para cada seña; en algunas, tal como la letra C, no se realiza una flexión completa de los dedos como si se hace en las letra S, no cumplir con este requisito implicaría lecturas erradas del grado de flexión de los dedos, notándose más en los dedos pulgar y meñique; por otro lado, cubrir las falanges mencionadas nos da mejores lecturas para una flexión total de cada dedo, para una flexión “intermedia” y para cuando están estirados, todo esto será representado por los niveles de voltaje de cada sensor Flex.



Figura N° 10: Disposición de sensores en el guante electrónico

Fuente: Elaboración propia

Los niveles de voltaje fueron interpretados por el módulo “Arduino nano” con microcontrolador ATMEGA 328p, para la programación de este se utilizaron las librerías "I2Cdev.h", "MPU6050.h" y "Wire.h", con 57600 baudios para la comunicación con MatLab, la disposición de los pines de entrada es mostrada en la Figura N° 11, así mismo los pines A4 y A5 están dispuestos para comunicación I2C (para el MPU6050) según el datasheet; los pines A0, A1, A2, A3 y A7 fueron destinados entonces para el sensor de cada dedo de la mano.

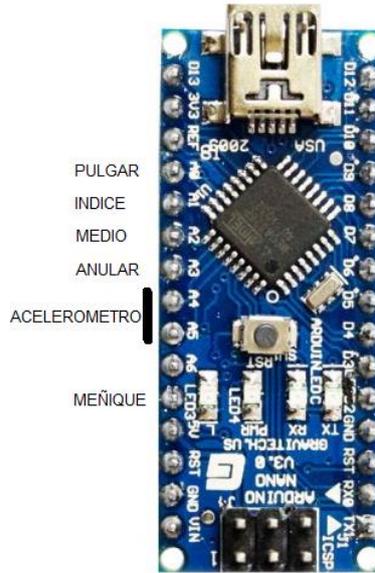


Figura N° 11: Disposición de pines en Arduino Nano

Fuente: (Elektorstore, s.f.)

En la programación del software MatLab la lectura de los datos escritos por Arduino en el puerto serial también posee un BaudRate de 57600 baudios, todo esto es estipulado en el comando “serial()” del software matlab, a continuación la apertura del puerto se realiza mediante el comando “fopen”.

Para evitar errores, se revisa periódicamente que la entrada de datos tenga un tamaño de 7x1, de manera que formen una matriz como la que se presenta a continuación en la expresión numérica (4):

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \end{bmatrix} \quad (4)$$

En la cual los valores desde a1 hasta a5 indican los niveles de voltaje de los sensores Flex y los valores a6 y a7 indican los valores del sensor MPU6050, siendo respectivamente los ejes X e Y. Cada una de estas matrices 7x1 simbolizan una muestra en concreto, a cada individuo le corresponden 200 muestras, las cuales se agrupan por consiguiente en matrices de 7x200 como se muestra a continuación en la expresión (5):

$$\begin{bmatrix} a_{1\ 1} & a_{1\ 2} & \cdots & a_{1\ 200} \\ a_{2\ 1} & a_{2\ 2} & \cdots & a_{2\ 200} \\ a_{3\ 1} & a_{3\ 2} & \cdots & a_{3\ 200} \\ a_{4\ 1} & a_{4\ 2} & \cdots & a_{4\ 200} \\ a_{5\ 1} & a_{5\ 2} & \cdots & a_{5\ 200} \\ a_{6\ 1} & a_{6\ 2} & \cdots & a_{6\ 200} \\ a_{7\ 1} & a_{7\ 2} & \cdots & a_{7\ 200} \end{bmatrix} \quad (5)$$

El proceso se repite para 3 individuos y para las 24 señas, de manera que se pueda tener al final una base de datos de 600 muestras para cada seña.

Se debe mencionar que debido a que los sensores Flex son comerciales, los valores de resistencia no son los mismos y mantienen una ligera diferencia que es notoria en los niveles de voltaje en los cuales trabajamos, para mantener un orden visual, se le da un ajuste numérico, de manera que los valores estén cercanos a la unidad al estar los dedos estirados, experimentalmente se pudo apreciar que al flexionar totalmente los dedos, los voltajes se encuentran por debajo de los 0.8 V, tal y como se puede apreciar en las Figuras N° 12 y 13 en las que se muestran algunos datos obtenidos con la mano estirada (ESTT) y con la mano cerrada (seña equivalente a la letra S del alfabeto dactilológico) respectivamente.

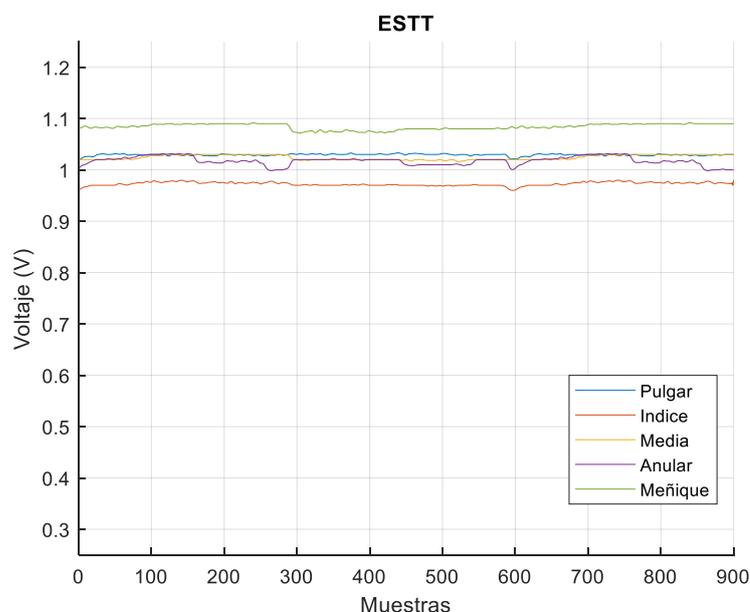


Figura N° 12: Voltajes con los dedos extendidos.

Fuente: Elaboración propia

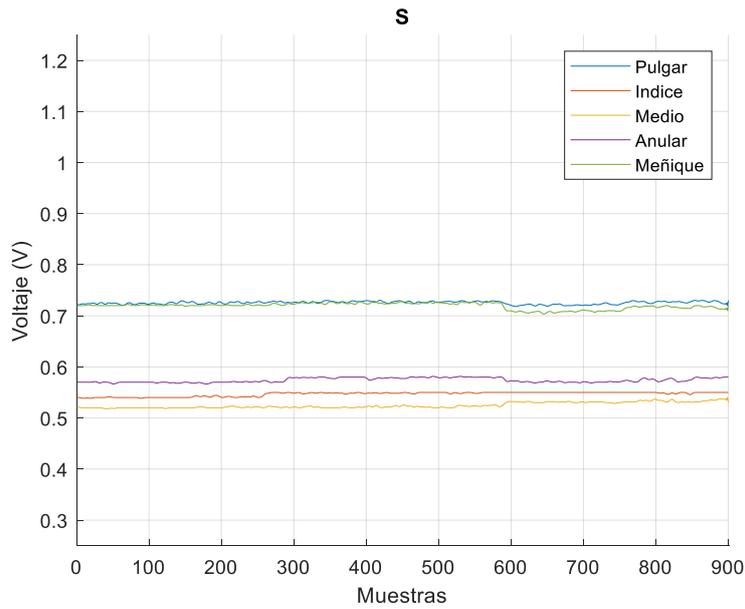
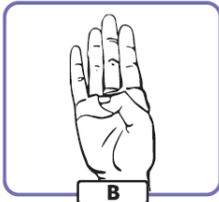


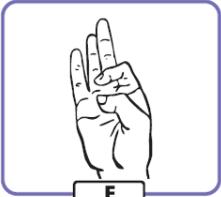
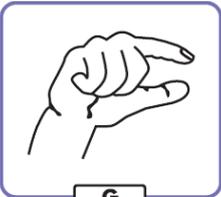
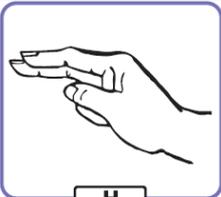
Figura N° 13: Valores de voltaje de los dedos contraídos.

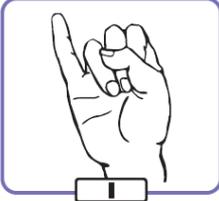
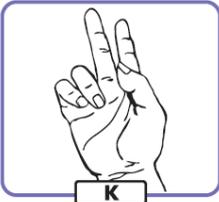
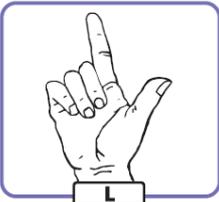
Fuente: Elaboración propia.

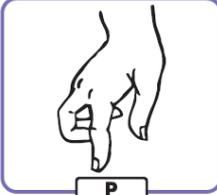
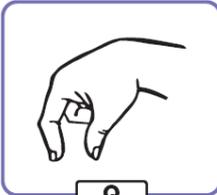
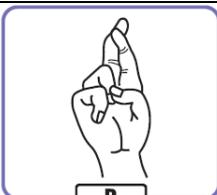
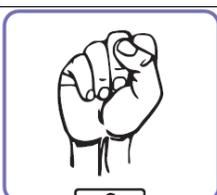
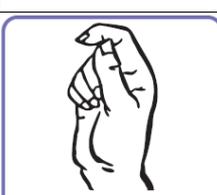
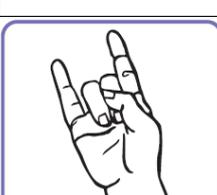
Por la cantidad de datos recopilados, se presenta a continuación en la Tabla 1 un resumen de los datos obtenidos por cada seña realizada, mostrándose voltajes máximos y mínimos que cada sensor flex posee en cada dedo respectivo.

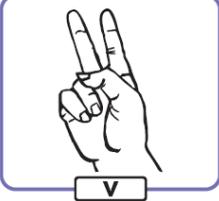
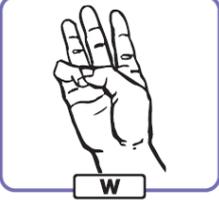
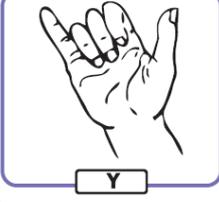
Tabla 1: Lectura en voltajes de sensores Flex

SEÑAS	PULGAR	ÍNDICE	MEDIO	ANULAR	MEÑIQUE
 A 1	0.9671 ~ 1.0109	0.6900 ~ 0.7168	0.5185 ~ 0.5506	0.5385 ~ 0.5612	0.6817 ~ 0.7126
 B 2	0.7997 ~ 0.8512	0.9900 ~ 1.0341	0.9864 ~ 1.0315	1.0902 ~ 1.1600	0.8900 ~ 0.9400

 <p>C</p> <p>3</p>	0.9834 ~ 1.0700	0.7581 ~ 0.8200	0.6200 ~ 0.6767	0.8063 ~ 0.8551	0.7323 ~ 0.8000
 <p>D</p> <p>4</p>	0.8506 ~ 0.9601	1.0622 ~ 1.1422	0.4676 ~ 0.5516	0.6504 ~ 0.7422	0.5018 ~ 0.6243
 <p>E</p> <p>5</p>	0.7500 ~ 0.7709	0.5700 ~ 0.6114	0.4300 ~ 0.4598	0.6300 ~ 0.6643	0.4965 ~ 0.5300
 <p>F</p> <p>6</p>	0.7744 ~ 0.8000	0.6016 ~ 0.6229	0.8400 ~ 0.8815	0.9600 ~ 1.0400	0.8372 ~ 0.8800
 <p>G</p> <p>7</p>	0.9300 ~ 0.9600	0.8885 ~ 0.9222	0.4592 ~ 0.5000	0.5625 ~ 0.5996	0.6447 ~ 0.6715
 <p>H</p> <p>8</p>	1.0486 ~ 1.0641	1.1210 ~ 1.1700	0.9327 ~ 0.9700	0.6963 ~ 0.7200	0.5271 ~ 0.5600

 <p style="text-align: center;">I</p> <p style="text-align: center;">9</p>	0.7406 ~ 0.7646	0.6400 ~ 0.6969	0.4986 ~ 0.5355	0.5925 ~ 0.6230	1.0385 ~ 1.0600
 <p style="text-align: center;">K</p> <p style="text-align: center;">10</p>	0.8482 ~ 0.8766	0.9209 ~ 0.9523	1.0100 ~ 1.0231	0.6013 ~ 0.6229	0.6802 ~ 0.7220
 <p style="text-align: center;">L</p> <p style="text-align: center;">11</p>	0.9783 ~ 1.0097	0.9283 ~ 0.9600	0.5002 ~ 0.5600	0.4872 ~ 0.5300	0.7085 ~ 0.7600
 <p style="text-align: center;">M</p> <p style="text-align: center;">12</p>	0.7728 ~ 0.8100	0.6904 ~ 0.7200	0.6022 ~ 0.6315	0.8231 ~ 0.8400	0.7391 ~ 0.7703
 <p style="text-align: center;">N</p> <p style="text-align: center;">13</p>	0.7686 ~ 0.7902	0.7100 ~ 0.7474	0.8126 ~ 0.8444	0.4980 ~ 0.5619	0.6480 ~ 0.6716
 <p style="text-align: center;">O</p> <p style="text-align: center;">14</p>	0.7400 ~ 0.7900	0.5700 ~ 0.6400	0.6100 ~ 0.6600	0.6300 ~ 0.6828	0.6800 ~ 0.7515

 <p>P</p> <p>15</p>	0.9663 ~ 1.0100	0.8777 ~ 0.9100	0.7660 ~ 0.8000	0.4885 ~ 0.5312	0.5900 ~ 0.6315
 <p>Q</p> <p>16</p>	0.8987 ~ 0.9200	0.8800 ~ 0.9134	0.4846 ~ 0.5000	0.4984 ~ 0.5200	0.6585 ~ 0.6683
 <p>R</p> <p>17</p>	0.6689 ~ 0.7200	0.8201 ~ 0.8449	0.9359 ~ 0.9700	0.5530 ~ 0.6015	0.5676 ~ 0.6028
 <p>S</p> <p>18</p>	0.7182 ~ 0.7303	0.5385 ~ 0.5500	0.5181 ~ 0.5400	0.5664 ~ 0.5815	0.7035 ~ 0.7289
 <p>T</p> <p>19</p>	0.8513 ~ 0.8934	0.7485 ~ 0.7824	0.5621 ~ 0.5789	0.6161 ~ 0.6300	0.6406 ~ 0.6543
 <p>U</p> <p>20</p>	0.7402 ~ 0.7799	0.8700 ~ 0.9215	0.4900 ~ 0.5400	0.6471 ~ 0.6700	1.0393 ~ 1.1184

 V 21	0.7100 ~ 0.7388	0.8016 ~ 0.8713	0.9936 ~ 1.0200	0.5110 ~ 0.5568	0.5977 ~ 0.6507
 W 22	0.7278 ~ 0.7500	0.8386 ~ 0.8629	1.0154 ~ 1.0222	0.8897 ~ 0.9178	0.5785 ~ 0.5826
 X 23	0.7351 ~ 0.7700	0.7400 ~ 0.7952	0.5074 ~ 0.5179	0.5570 ~ 0.5862	0.6959 ~ 0.7100
 Y 24	0.9785 ~ 1.0269	0.6985 ~ 0.7119	0.6151 ~ 0.6300	0.5768 ~ 0.6021	0.9943 ~ 1.0269

De lo anteriormente presentado, se pueden apreciar algunas señas “críticas” entorno a la similitud que poseen, estas son:

- D – X
- C - O
- E – M - N-S
- K – H
- L – G - Q
- R - V

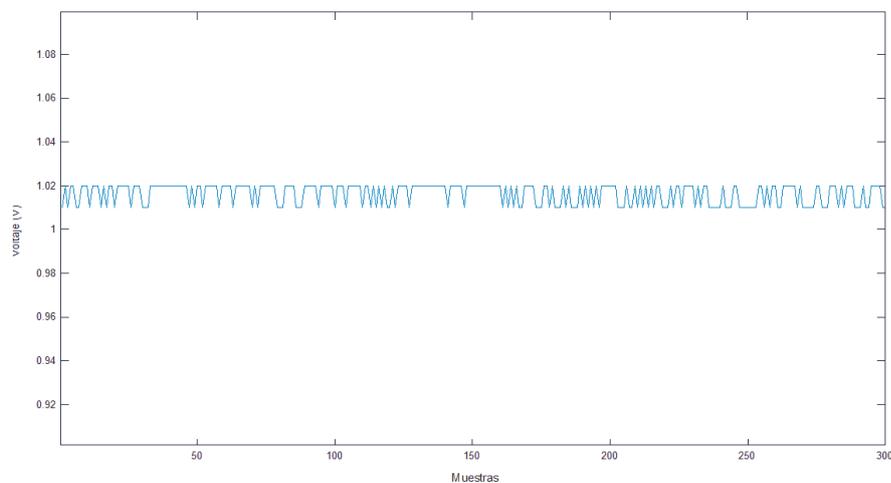
Un claro ejemplo son las señas de las letras C y O, que solo se diferencian en “cuanto” es que se curvan los dedos (mucho más cerrados al formar la segunda) y por tanto en valores de voltaje mayores; otro caso se puede apreciar en las letras G, L y Q, en las que los dedos pulgar e índice se encuentran estirados y poseen una ligera variación

entorno a la posición de la mano, paralela al eje X del sensor MPU en una, orientada a Y+ en otra y hacia Y- en la última. Se presentan a continuación en la Tabla 2 los valores obtenidos del sensor MPU para las distintas orientaciones descritas anteriormente, las que son críticas en el proceso de decisión de las redes neuronales.

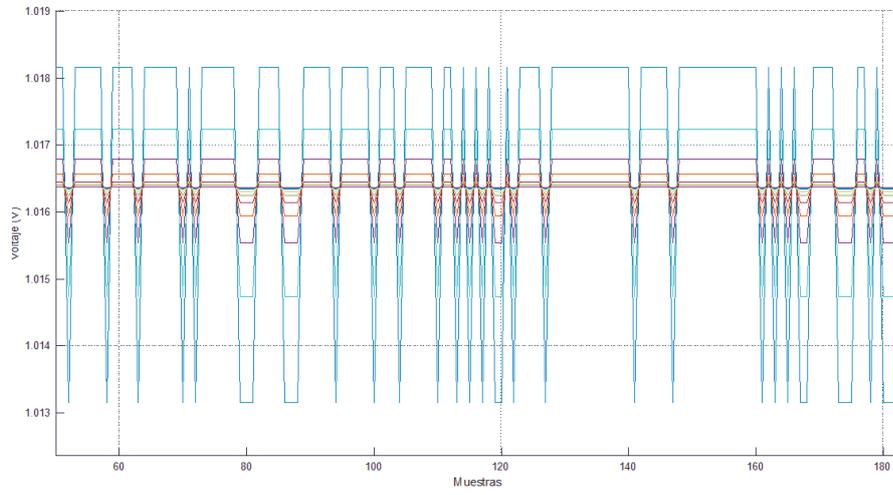
Tabla 2: Lectura de sensor MPU

Orientación	Eje X	Eje Y
X	29.8495 ~ 31.2982	-47.7389 ~ -45.2999
Y+	11.2813 ~ 17.0790	-76.9832 ~ -72.0087
Y-	17.8459 ~ 21.4277	48.7415 ~ 50.7634

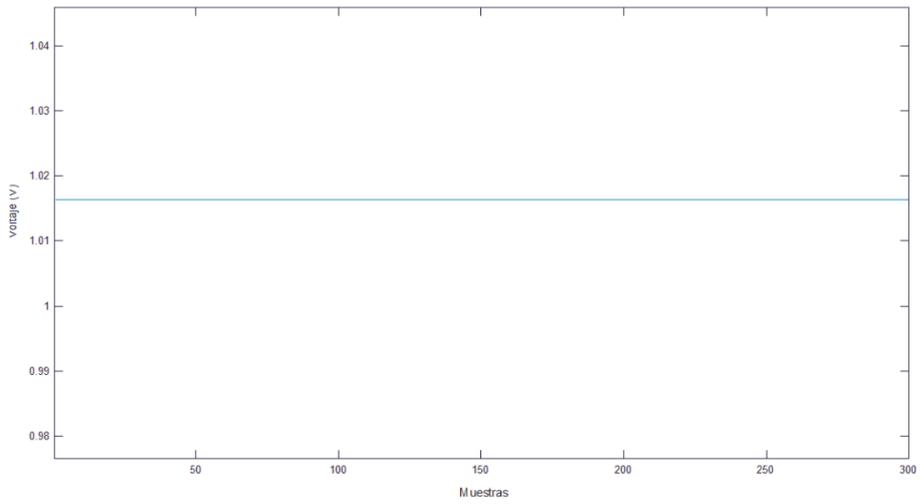
El preprocesamiento de las señas se encarga de revisar cada muestra y compararla con el resto de estas, identificando si existen picos producto de una mala lectura, además de instaurar un algoritmo de normalización de los datos para evitar una variación numérica que pudiera afectar a la red neuronal, de manera que, iteración a iteración, se va ajustando una media de todos los datos obtenidos, de la manera como se aprecia en la Figura N° 14 presentada a continuación en la cual se aplica este algoritmo sobre un conjunto de muestras de la letra A. En la figura a) se aprecia el conjunto de datos en un inicio, a continuación, en b) y con una vista aumentada, se aprecia el proceso de normalización de la señal y finalmente en c) se tiene el resultado final del preprocesamiento.



a)



b)



c)

Figura N° 14: Normalización de señal de entrada mediante el preprocesamiento.

Fuente: Elaboración propia.

3.3.2. Red Neuronal de Regresión Generalizada (GRNN)

3.3.2.1. Implementación computacional de GRNN

Para la implementación de esta red nos basamos en el capítulo 22 de la Guía de Usuario – Deep Learning del software MatLab, en donde se indica de manera explícita que para la creación de una GRNN se debe usar el comando “newgrnn”. (MathWorks, 2020) Este comando está definido por tres parámetros importantes indicados en la Tabla 3:

Tabla 3: Comando para diseño de red GR

newgrnn (P, T, SPREAD)	
P	Vector de entrada, es decir, los datos recabados en el proceso experimental.
T	Vector de clases u “objetivos”, en esta sección se deben estipular las distintas clases sobre las que recae cada dato de entrada de manera respectiva.
SPREAD	Con un valor estándar 1, este parámetro define la “suavidad” de la función de aproximación.

Para este caso, se implementaron 2 redes neuronales GR, una para que reconozca los datos provenientes de los dedos, y otra red que se encargue de los provenientes del sensor MPU.

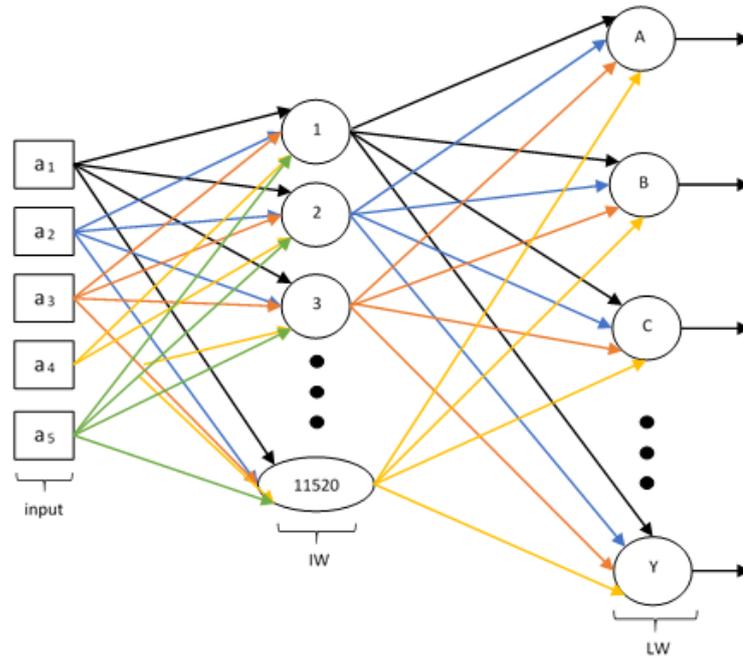


Figura N° 15: Diagrama de Red GR para señas.

Fuente: Elaboración propia.

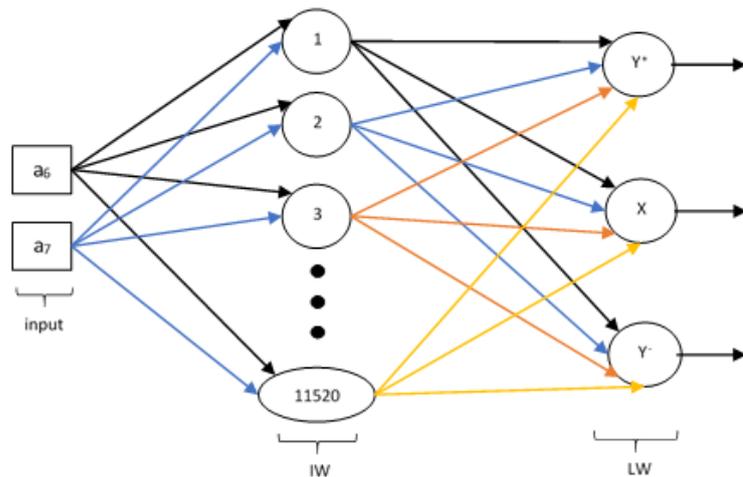


Figura N° 16: Diagrama de Red GR para sensor MPU.

Fuente: Elaboración propia.

En las Figuras N° 15 y 16, en la parte izquierda y como se mostró anteriormente, se encuentran las neuronas de entrada (con forma cuadrada) para ambas redes neuronales, la primera de 5 y la segunda de 2 neuronas de entrada; a continuación, se ven las neuronas ocultas IW, que son 11520 para la primera y segunda red que determinan las señas y la posición de la mano en base al sensor MPU respectivamente; y finalmente las neuronas de salida LW, 24 para las señas y 3 para el sensor MPU, que representan cada una de las orientaciones definidas como Y^+ , X e Y^- .

3.3.2.2. Entrenamiento y prueba básica de Red Neuronal

Para esta red neuronal en específico, el entrenamiento se realiza mediante el cálculo de pesos en el momento en que se diseña la red con el comando “lvqnet” especificado anteriormente; los vectores de entrenamiento son llamados “probe” para la red de señas y “MPU” para la red MPU y los vectores de clase son llamados “Tc5” y “Tcmpru” respectivamente, las pruebas básicas consisten en la aplicación del mismo vector de entrenamiento sobre las redes, dando como resultados “Ycg5” e “Ycgmpu”; lo ideal para esta prueba básica es que la red pueda clasificar su vector de entrenamiento sin errores; y en este caso particular, el resultado fue alcanzado por la red de señas y por la red del sensor MPU (con 11520 neuronas). Esto se puede apreciar al comparar los vectores target “Tc5” y “Tcmpru” con los resultados de la red neuronal “Ycg5” e “Ycgmpu”.

```
>> resultadog5 = Ycg5~=Tc5;
>> match_GR5 = find(resultadog5)
match_GR5 =
    1×0 empty double row vector

>> resultadogmpu = Ycgmpu~=Tcmpru;
>> match_Gmpu = find(resultadogmpu)
match_Gmpu =
    1×0 empty double row vector
```

En la previa rutina de programación mostrada, se puede apreciar que el resultado obtenido en ambos casos es un resultado nulo, un vector vacío, con lo que certificamos que la red no ha tenido equivocaciones al clasificar a su vector de entrenamiento.

3.3.3. Red Neuronal de Aprendizaje por Cuantificación Vectorial (LVQNN)

3.3.3.1. Implementación computacional de LVQNN

Para la implementación de esta red se utilizó el ejemplo práctico con la Red LVQ desarrollado en el Cuaderno de Ingeniería: Simulación de redes neuronales con MatLab (Huamaní, 2008), y también del capítulo 23 de la Guía de Usuario – Deep Learning del software MatLab, en donde se indica de manera explícita que para la creación de una LVQNN se debe usar el comando “lvqnet”. (MathWorks, 2020) Este comando está definido por tres parámetros indicados en la Tabla 4:

Tabla 4: Comando para diseño de red LVQ

lvqnet (hiddenSize, lvqLR, lvqLF)	
hiddenSize	En esta sección se debe estipular el número de neuronas de la capa oculta de la red neuronal, el número de grupos o clases está determinado por este parámetro.
LR	Aquí se estipula la tasa de aprendizaje de la red neuronal, su valor estándar es de 0.01.
LF	Aquí se selecciona la función de aprendizaje, una red LVQ posee 2 tipos de aprendizajes, lvq1 y lvq2 (el segundo se aplica únicamente después de haberse entrenado la red con lvq1).

Fuente: Elaboración propia.

De manera análoga, se implementaron 2 redes neuronales LVQ, una para que reconozca los datos provenientes de los dedos, y otra red encargada de los provenientes del sensor MPU. En las Figuras N° 17 y 18, en la parte izquierda y como se mostró anteriormente, se encuentran las neuronas de entrada (con forma cuadrada) para ambas redes neuronales, la primera de 5 y la segunda de 2 neuronas

de entrada; a continuación, se ven las neuronas ocultas IW, que son 96 para la primera red que determina las señas y 9 para la que determina la posición de la mano en base al sensor MPU; y finalmente las neuronas de salida LW, 24 para las señas y 3 para el sensor MPU.

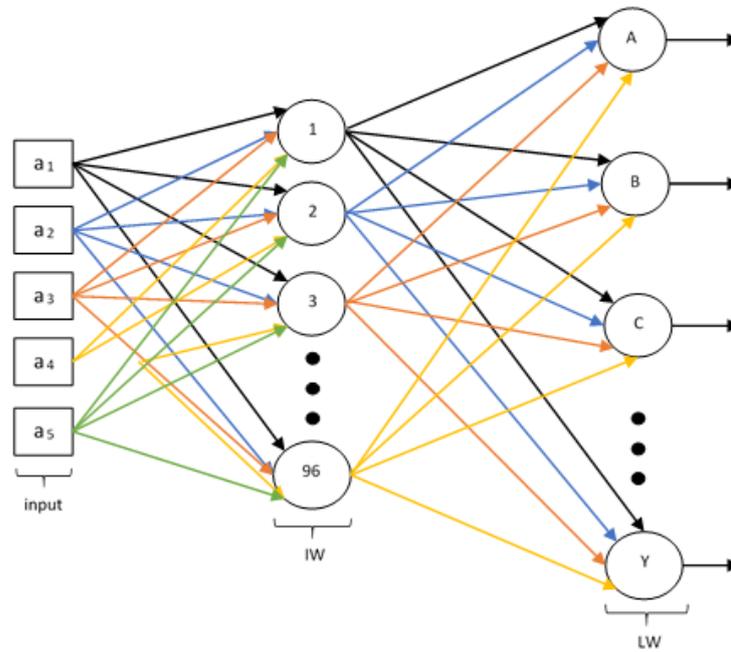


Figura N° 17: Diagrama de Red LVQ para señas.

Fuente: Elaboración propia.

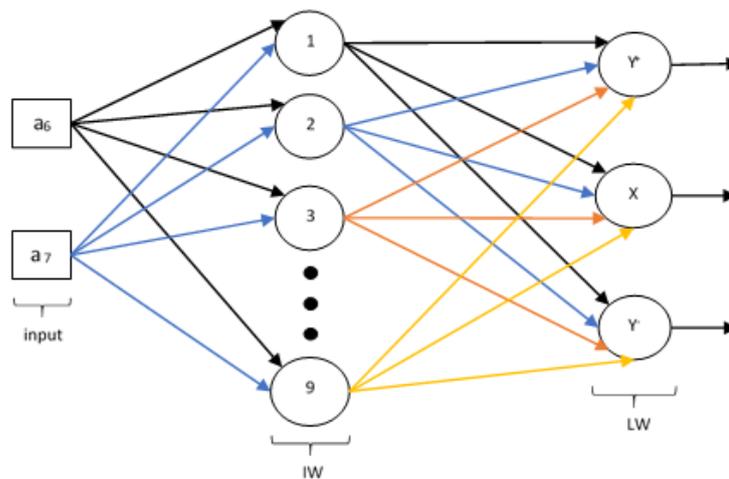


Figura N° 18: Diagrama de Red LVQ para sensor MPU.

Fuente: Elaboración propia.

3.3.3.2. Entrenamiento y prueba básica de Red Neuronal

Para el entrenamiento de esta red neuronal se deben tener en cuenta algunos puntos importantes, esta red, comparada a la anterior, no realiza un proceso de cálculo como tal si no un proceso de “aprendizaje”, en donde va actualizando sus pesos iteración a iteración (o “epoch a epoch” como lo denomina MatLab), el software posee un límite de iteraciones o “epochs” al realizar el entrenamiento de redes neuronales de manera estándar, en caso se necesiten más, este número se puede cambiar antes de realizar el entrenamiento mediante el comando señalado a continuación en la expresión (6):

```
red.trainParam.epochs
```

 (6)

Como lo fue previamente mencionado, el proceso de aprendizaje o entrenamiento consiste en la actualización de los pesos neuronales, de manera que los vectores de entrada “alcancen” el objetivo o clase definido en el vector target; para realizar este aprendizaje se utiliza el comando “train”, detallado en la expresión (7), que tal como se indica, requiere como parámetros de entrada el nombre de la red a entrenar (“red”), el vector de entrada (“input”) y el vector de clases (“target”).

```
train(red, input, target)
```

 (7)

Los vectores de entrenamiento son llamados “probe” para la red de señas y “MPU” para la red MPU; y los vectores de clase “Tc5” y “TcmPU” respectivamente. Las pruebas básicas de esta red consisten en la aplicación del mismo vector de entrenamiento sobre las redes, dando como resultado “Yc5” e “YcmPU”; el resultado ideal para esta prueba es que la red pueda clasificar su vector de entrenamiento sin errores, para este caso, la red MPU alcanzó el valor ideal, sin embargo en la red de Señas se alcanzó a clasificar erróneamente 1440 datos de los 11520 ingresados, lo cual estaba dentro de lo previsto por algunas señas que se consideraban como críticas y que necesitaban de los datos del sensor MPU para su clasificación. Se presenta parte de la rutina de programación:

```
>> resultadompu = Ycmpu~=Tcmpu;
>> match_LVQmpu = find(resultadompu)
match_LVQmpu =
1x0 empty double row vector
```

```
>> resultado5 = Yc5~=Tc5;
>> match_LVQ5 = find(resultado5);
>> size(match_LVQ5)
ans =
1      1440
```

A continuación se presentan las gráficas del entrenamiento realizado en la red de señas; en la Figura N° 19 se muestra la variación el error en cada una de las iteraciones de la red neuronal, el menor error se dio en el epoch 19, con un valor de 0.010417 y a partir del cual no existieron variaciones en el error, siendo este el mínimo al cual podía llegar la red, esto se aprecia tambien en la Figura N° 20, donde se muestra un diagrama de la red implementada, los epoch realizados, el tiempo de entrenamiento y el error de 0.0104.

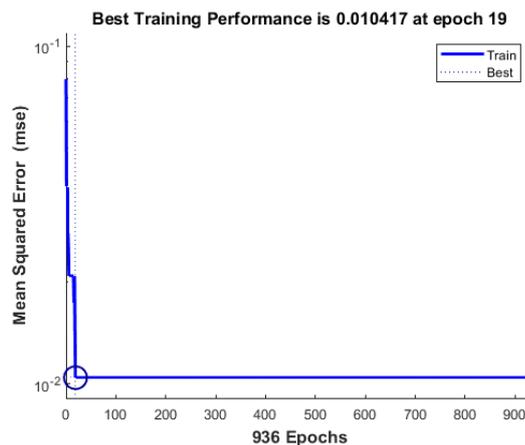


Figura N° 19: Grafica de entrenamiento Error vs Iteraciones

Fuente: Elaboración propia.

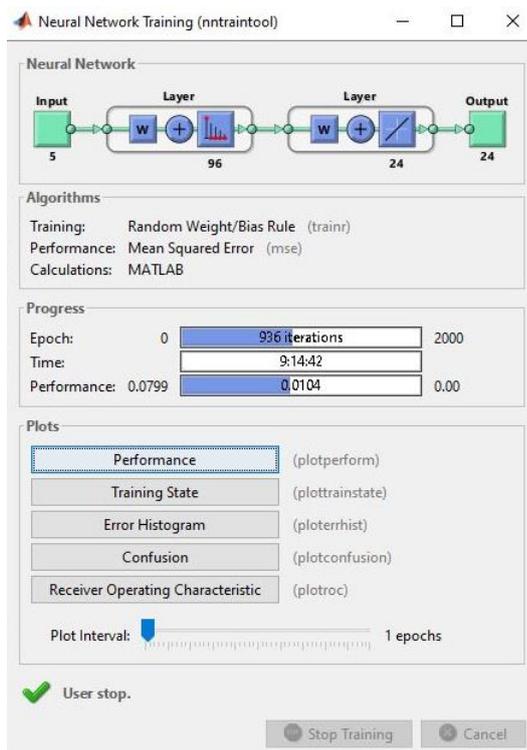


Figura N° 20: Herramienta de entrenamiento del software MatLab.

Fuente: Elaboración propia.

CAPÍTULO IV: PRUEBAS Y RESULTADOS

En el siguiente capítulo se describen los diferentes resultados obtenidos al ingresar los vectores de prueba a las redes neuronales previamente entrenadas, obteniendo con los resultados las tablas de contingencia y los valores de precisión y recall de cada red implementada según la técnica de Clustering aplicada.

Los resultados para cada tabla de contingencia fueron obtenidos mediante el software Matlab ya que nos brinda el comando presentado en la expresión (8) que, con los parámetros de entrada “target” (vector de clases definidas por el usuario”) y “resultado” (resultados finales de las redes neuronales), construye la tabla de contingencia:

$$\text{plotconfusion (target , resultado);} \quad (8)$$

Para los valores de precisión y recall se utilizaron las expresiones matemáticas (1) y (2) (pág. 27 y 28) de manera respectiva. En adición a ello, son presentados también resultados de la aplicación de ruido a las señales, utilizando los valores de la precisión y recall de las redes para los casos descritos, de manera que se pueda tener un panorama de la actuación de estas.

4.1. Redes Neuronales de Regresión Generalizada: GRNN

Se poseen 3 redes distintas, una encargada de clasificar los datos obtenidos de los 5 dedos de la mano (Red de señas), otra red encargada de clasificar los datos obtenidos del sensor MPU (Red MPU) y finalmente una red que combina los resultados de las redes previas (Red Conjugada), de manera que pueda ayudar en el discernimiento de la categoría correcta en caso existan similitudes.

4.1.1. Red de Señas de Regresión Generalizada

4.1.1.1. Evaluación de Tablas de Contingencia

Se presenta en la figura N° 21 la tabla de contingencia de esta red neuronal, donde se puede apreciar que la red logró clasificar de manera acertada, las 120 muestras de cada seña. Obteniendo una cantidad nula tanto de falsos positivos como falsos negativos para cada una de las categorías establecidas.

4.1.2. Red MPU de Regresión Generalizada

4.1.2.1. Evaluación de Tablas de Contingencia

La Tabla de contingencia presentada en la Figura N° 22 presenta la clasificación de la Red entorno a los datos MPU obtenidos de cada seña, se observa que no existieron errores en la clasificación de los datos MPU en la clase que determinaba la orientación del sensor.

1	2040 70.8%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	600 20.8%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	240 8.3%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
	1	2	3	
	Target Class			

Figura N° 22: Tabla de contingencias para red MPU GR.

Fuente: Elaboración propia.

4.1.2.2. Evaluación de Precisión y Recall

De igual manera, para cada clase correctamente clasificada, los valores de precisión y recall son iguales a la unidad, según lo establecido en las ecuaciones (1) y (2), se comprueban entonces estos valores tomando como ejemplo los valores de la categoría 1, detallado en las expresiones matemáticas (11) y (12):

$$precision = \frac{2040}{0 + 2040} = 1 \quad (11)$$

$$recall = \frac{2040}{0 + 2040} = 1 \quad (12)$$

Estos valores implican una capacidad del 100% en la clasificación de los datos obtenidos de los guantes según la seña que se esté gesticulando con estos.

4.2. Redes Neuronales de Aprendizaje por Cuantificación Vectorial: LVQNN

Para esta técnica de clustering se dispone de 3 redes distintas, una encargada de clasificar los datos obtenidos de los 5 dedos de la mano (Red de señas), otra red encargada de clasificar los datos obtenidos del sensor MPU (Red MPU) y finalmente una red que combina los resultados de las redes previas (Red Conjugada), de manera que pueda ayudar en el discernimiento de la categoría correcta en caso existan similitudes.

4.2.1. Red de Señas de Aprendizaje por Cuantificación Vectorial

4.2.1.1. Evaluación de Tablas de Contingencia

La tabla de contingencias mostrada en la Figura N° 24 expone errores en la clasificación de 3 señas en particular:

Confusion Matrix

Output Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24		
1	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%	
2	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
3	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
4	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
5	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
6	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
7	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
8	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
9	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
10	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
11	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
12	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	100%
13	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	100%
14	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	100%
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	100%
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	100%
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	100%
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	100%
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	100%
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	100%
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	100%
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	100%
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	100%
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	100%
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	

Target Class

Figura N° 24: Tabla de contingencias de la red de señas LVQ.

Fuente: Elaboración propia.

- 120 muestras de la letra G (clase 7) son consideradas como parte de la letra T (clase 19).
- 120 muestras de la letra Q (clase 16) son consideradas como parte de la letra X (clase 23).
- 120 muestras de la letra S (clase 18) son consideradas como parte de la letra O (clase 14).

4.2.1.2. Evaluación de Precisión y Recall

De lo anteriormente expuesto, notamos que para las categorías 14, 19 y 23 se presentan un $f_p = 120$; para las categorías 7, 16 y 18 un $f_n = 120$. Con lo cual obtenemos para cada categoría los valores presentados en la Tabla 5.

Tabla 5: Valores de RECALL y PRECISIÓN para red de señas LVQ

LETRA	PRECISIÓN	RECALL
A	1	1
B	1	1
C	1	1
D	1	1
E	1	1
F	1	1
G	1	0
H	1	1
I	1	1
K	1	1
L	1	1
M	1	1
N	1	1
O	0.5	1
P	1	1
Q	1	0
R	1	1

S	1	0
T	0.5	1
U	1	1
V	1	1
W	1	1
X	0.5	1
Y	1	1

4.2.2. Red MPU de Aprendizaje por Cuantificación Vectorial

4.2.2.1. Evaluación de Tablas de Contingencia

De manera análoga, se presenta a continuación en la Figura N° 25 los resultados de clasificación de los datos del sensor MPU en la tabla de contingencias respectiva; en ella se aprecia que los datos son clasificados de manera correcta.



Figura N° 25: Tabla de contingencia para red MPU LVQ.

Fuente: Elaboración propia.

4.2.2.2. Evaluación de Precisión y Recall

Los valores para cada categoría previamente mostrada en la tabla de contingencia son iguales a la unidad como se ha ido mostrando anteriormente para cada categoría clasificada de manera correcta, de manera análoga, se tomó de

ejemplo esta vez a la categoría 2 con 600 elementos, obteniéndose los resultados de las expresiones (15) y (16).

$$precision = \frac{600}{0 + 600} = 1 \tag{15}$$

$$recall = \frac{600}{0 + 600} = 1 \tag{16}$$

4.2.3. Red Conjugada de Aprendizaje por Cuantificación Vectorial

4.2.3.1. Evaluación de Tablas de Contingencia

Finalmente se presenta en la Figura N° 26 la tabla de contingencia perteneciente a la red neuronal encargada de clasificar las señas en función de los datos obtenidos de la Red SEÑAS y de la Red MPU. En esta ocasión, al utilizar únicamente los datos de estas redes, pese al continuo entrenamiento se logró clasificar de manera correcta una sola categoría.

Confusion Matrix

Output Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	600	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Target Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24

Figura N° 26: Tabla de contingencia Red Conjugada LVQ.

Fuente: Elaboración propia.

4.2.3.2. Evaluación de Precisión y Recall

Para la tabla de contingencia previamente mostrada se obtienen los siguientes valores para cada una de las categorías según lo establecido en la Tabla 6. En donde debido a la no existencia de datos clasificados correctamente ni de falsos positivos, excepto en la categoría 1, correspondiente a la letra A, donde se acumulan todos los datos a los que es sometida la red neuronal.

Tabla 6: Valores de RECALL y PRECISIÓN para Red Conjugada LVQ

LETRA	PRECISIÓN	RECALL
A	0.0417	1
B	∅	0
C	∅	0
D	∅	0
E	∅	0
F	∅	0
G	∅	0
H	∅	0
I	∅	0
K	∅	0
L	∅	0
M	∅	0
N	∅	0
O	∅	0
P	∅	0
Q	∅	0
R	∅	0
S	∅	0
T	∅	0
U	∅	0
V	∅	0

W	≠	0
X	≠	0
Y	≠	0

4.3. Análisis y comparación de datos

En esta sección se realiza un análisis de los datos obtenidos previamente en cada red neuronal de acuerdo con los 3 parámetros establecidos como lo son la tabla de contingencia, la precisión y el recall.

De igual manera, y en consecuencia al 3^{er} ítem mencionado en la sección 2.3.2.1., donde se menciona que “la medida debe ser un único número, de manera que pueda ser transmitida de manera simple e inmediatamente perceptible.” (Walde, 2003), frente a la necesidad de comparar la actuación de todas estas redes, se utiliza la “magnitud-F” (F-measure) utilizada para obtener un valor numérico único que relaciona los valores obtenidos de recall y precisión en estudios como los de Csoka y Vasileios. La ecuación propuesta para esta medida es obtenida de la expresión matemática (3) con el valor $\beta = 1$, en la expresión numérica (17)

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \quad (17)$$

4.3.1. Tabla de contingencia

Las tablas de contingencia nos pueden dar una idea gráfica entorno a el desempeño de la red neuronal según los datos clasificados correctamente y los que no han logrado el cometido.

De las Figuras N°21 y 24 anteriormente mostradas, podemos verificar que entorno a las señas, la red GRNN tiene una mayor precisión en la clasificación, pudiendo clasificar de manera correcta las señas en su categoría sin tener en cuenta las señas “criticas” que definimos en un inicio; por otro lado, la red LVQ si se ve afectada y es un poco más susceptible a esta similitud en los datos. Por otro lado, las redes MPU tanto para la red

GR como para la red LVQ llegan a clasificar de manera idónea cada uno de los datos, como se puede ver en las Figuras N°22 y 25.

Finalmente, en las Figuras N° 23 y 26, las cuales describen la actuación de clasificación de la Red Conjugada GR y LVQ respectivamente en función de los datos de las anteriores redes neuronales, se puede apreciar una gran diferencia; la red GR como era de esperarse, logra clasificar correctamente todos los datos a los cuales fue sometida, en contraposición a la segunda, la cual, solo puede clasificar una categoría, en la cual son acumulados los falsos positivos y donde los falsos negativos son nulos.

4.3.2. Precisión

Como se mencionó anteriormente y de la ecuación (1) descrita en el Capítulo II, la “precisión” es directamente proporcional a la cantidad de datos correctamente clasificados por la red e inversamente proporcional al total de datos clasificados por la red en una categoría específica.

La “Red de señas” bajo la técnica GR no tuvo inconvenientes al momento de realizar la clasificación, por lo cual el valor de precisión en cada categoría es igual a la unidad (expresión numérica (9)), de igual manera con la red de técnica LVQ, con excepción de 3 categorías específicas (Tabla 5). Un caso distinto se aprecia al verificar la precisión de las “Redes Conjugadas” de cada técnica de clustering, donde se adquiere un valor igual a la unidad en cada categoría para la técnica GR (expresión numérica (13)), sin embargo, en la LVQ se pueden apreciar valores no determinados debido a la no existencia de elementos correctamente clasificados por la red neuronal, se expresa esta cantidad con el símbolo \nexists de no existencia (Tabla 6), en esta misma red, para la categoría “A” la red clasifica correctamente los 120 elementos pertenecientes a la misma, pero obtiene un valor de 0.0417 de precisión debido a la acumulación de falsos positivos. Se presenta a continuación en la Tabla N° 7 un resumen de los valores expuestos anteriormente.

Tabla 7: Comparación de valores de precisión por letra.

LETRA	Red de Señas		Red Conjugada	
	Precisión GR	Precisión LVQ	Precisión GR	Precisión LVQ
A	1	1	1	0.0417
B	1	1	1	∅
C	1	1	1	∅
D	1	1	1	∅
E	1	1	1	∅
F	1	1	1	∅
G	1	1	1	∅
H	1	1	1	∅
I	1	1	1	∅
K	1	1	1	∅
L	1	1	1	∅
M	1	1	1	∅
N	1	1	1	∅
O	1	0.5	1	∅
P	1	1	1	∅
Q	1	1	1	∅
R	1	1	1	∅
S	1	1	1	∅
T	1	0.5	1	∅
U	1	1	1	∅
V	1	1	1	∅
W	1	1	1	∅
X	1	0.5	1	∅
Y	1	1	1	∅
PROM.	1	0.9375	1	1.7375x10 ⁻³

4.3.3. Recall

De igual manera y como se mencionó anteriormente y de la ecuación (2) descrita en el Capítulo II, el recall es directamente proporcional a la cantidad de datos correctamente clasificados por la red e inversamente proporcional al total de datos que debería existir en una categoría específica.

Para las redes de señas existe la misma diferencia en los valores vista al analizar la precisión, con la diferencia de radicar en las categorías en las que deberían residir los datos de manera idónea.

Así mismo, la “Red de señas” de técnica GR fue exacta al momento de realizar la clasificación, por lo cual el valor de recall en cada categoría es igual a la unidad (expresión matemática (10)); por otro lado, en la técnica LVQ, bajo lo descrito en la Tabla 5, se encuentran valores nulos de este parametro en las categorías que no fueron correctamente clasificadas. Al verificar el valor de recall de la “Red Conjugada” de cada técnica de clustering, este es igual a la unidad en cada categoría para la técnica GR (expresión numérica (14)), mientras que en la perteneciente a la técnica LVQ (Tabla 6), los valores son nulos debido a la no existencia de valores correctamente clasificados, pero con existencia de falsos negativos, residentes en la categoría 1, correspondiente a la letra A.

Se presenta a continuación en la Tabla N° 8 un resumen de los valores expuestos anteriormente.

Tabla 8: Comparación de valores de recall por letra.

LETRA	Red de Señas		Red Conjugada	
	Recall GR	Recall LVQ	Recall GR	Recall LVQ
A	1	1	1	1
B	1	1	1	0
C	1	1	1	0
D	1	1	1	0
E	1	1	1	0
F	1	1	1	0
G	1	0	1	0

H	1	1	1	0
I	1	1	1	0
K	1	1	1	0
L	1	1	1	0
M	1	1	1	0
N	1	1	1	0
O	1	1	1	0
P	1	1	1	0
Q	1	0	1	0
R	1	1	1	0
S	1	0	1	0
T	1	1	1	0
U	1	1	1	0
V	1	1	1	0
W	1	1	1	0
X	1	1	1	0
Y	1	1	1	0
PROM.	1	0.9583	1	0.0417

4.3.4. Magnitud F

Esta magnitud nos sirve para poder determinar numéricamente la relación existente entre la precisión y el recall de una red neuronal, de manera que se pueda definir numéricamente y con un solo parámetro el desempeño de una red neuronal para una función determinada.

Para las redes neuronales GR se tomaron los valores promedio obtenidos de las expresiones numéricas desde la (9) hasta la (14), en donde el valor obtenido puede ser aplicado en todas las demás categorías, obteniendo un valor promedio igual a la unidad como promedio, y ciñéndonos a lo especificado en la expresión matemática (17) de la magnitud-F, obtenemos también un valor igual a la unidad. Del mismo modo, para las redes neuronales LVQ, se tomaron los valores promedio de cada red neuronal, de la

Tabla 5 para la red de señas, expresiones numéricas (15) y (16) para la red MPU y de la tabla 6 para la red conjugada para poder obtener el valor de la magnitud F de cada una de estas redes implementadas.

Se presenta a continuación en la Tabla 9 los valores promedio de precisión y recall para cada red neuronal (señas, MPU y conjugada) bajo las técnicas de clustering GR y LVQ junto con su magnitud-F característica.

Tabla 9. Magnitud F característica

Redes de Regresión Generalizada						
	Red de Señas		Red MPU		Red Conjugada	
	Precisión	Recall	Precisión	Recall	Precisión	Recall
	1	1	1	1	1	1
Magnitud-F	1		1		1	
Redes de Aprendizaje por Cuantificación Vectorial						
	Red de Señas		Red MPU		Red Conjugada	
	Precisión	Recall	Precisión	Recall	Precisión	Recall
	0.9375	0.9583	1	1	1.7375×10^{-3}	0.04167
Magnitud-F	0.9677		1		3.3359×10^{-3}	

4.4. Aplicación de Red Conjugada GR en los datos LVQ

4.4.1. Tabla de contingencia

Como una posible solución dentro de lo estudiado, se aplicó una Red neuronal GR como sustitución a la Red Neuronal Conjugada LVQ, los resultados de esta operación se ven reflejados en la Figura N° 27, que en contraste con lo presentado en la Figura N° 26 presenta una mejora sustancial, logrando de esta manera la clasificación correcta de las señas previamente mencionadas como errores y que requerían del MPU para su correcta clasificación.

Confusion Matrix

Output Class	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
1	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	120

Target Class

Figura N° 27: Tabla de contingencia Red Conjugada GR en LVQ.

Fuente: Elaboración propia.

4.4.2. Evaluación de Precisión y Recall

De los resultados obtenidos anteriormente, se obtienen valores iguales a la unidad para cada categoría que ha sido clasificada de manera correcta, de la manera como se presenta en las expresiones (18) y (19).

$$precision = \frac{120}{0 + 120} = 1 \tag{18}$$

$$recall = \frac{120}{0 + 120} = 1 \tag{19}$$

4.5. Aplicación de ruido a las redes neuronales

A continuación, y para tener una mejor apreciación del desempeño de nuestras redes neuronales, se agregó un ruido controlado a las señales existentes en proporciones del 0.1, 1, 5 y 10% en el software MatLab mediante el comando presentado en la expresión (20):

$$ruido = randi([i_{min} \ i_{max}], SZ_1, SZ_2) / percentage ; \quad (20)$$

Con el cual se producirán números aleatorio en el rango establecido $[i_{min} \ i_{max}]$, en un vector de dimensiones $|SZ_1 \times SZ_2|$, así mismo para obtener los porcentajes establecidos se debe considerar la proporción (21):

$$Nivel \ de \ ruido \ \% = \frac{i_{max}}{percentage} * 100 \% \quad (21)$$

Para los resultados obtenidos de la experimentación se presentan las tablas resumen de los valores de precisión y recall de cada red neuronal, así como también los valores promedio de estos parámetros (Thalange, 2015) y la magnitud-F (Vasileios, 1993) que relacione estos valores, según la expresión matemática (17). Es pertinente señalar que al aplicar ruido a las señales MPU no se encontraron errores y la clasificación fue idónea en cada una de las redes anteriormente mencionadas, por lo que sus resultados no serán mencionados en esta sección al no tener variaciones; además, se mantiene en este punto la Red conjugada GR aplicada en la sección 4.4 para las redes de señas y MPU con técnica LVQ.

Se presenta entonces en la Tabla N° 10 los valores de precisión y recall de la red de señas y de la red conjugada al aplicarles la señal con 0.1% de ruido en cada técnica de clustering. Los resultados nos dejan ver que con este porcentaje de ruido las redes neuronales GR permanecen inalterables en el proceso de clasificación; contrapuesto a los resultados arrojados por las redes LVQ, que a este bajo nivel de ruido se ven afectadas en su tarea. En este caso se pueden apreciar errores de clasificación de la red de señas, que se ven mejorados al agregar correctamente los datos obtenidos en el MPU, esto se aprecia en la salida de la Red Conjugada.

Tabla 10: Aplicación de ruido al 0.1% sobre redes neuronales

Letras	Red Señas				Red Conjugada			
	Red GR		Red LVQ		Red GR		Red LVQ	
	Precisión	Recall	Precisión	Recall	Precisión	Recall	Precisión	Recall
A	1	1	1	1	1	1	1	1
B	1	1	1	1	1	1	1	1
C	1	1	1	1	1	1	1	1
D	1	1	1	1	1	1	1	1
E	1	1	1	1	1	1	1	1
F	1	1	1	1	1	1	1	1
G	1	1	∅	0	1	1	1	0.5083
H	1	1	1	1	1	1	1	1
I	1	1	1	1	1	1	1	1
K	1	1	1	1	1	1	1	1
L	1	1	0.4211	1	1	1	0.6704	1
M	1	1	1	1	1	1	1	1
N	1	1	1	1	1	1	1	1
O	1	1	0.5021	1	1	1	1	1
P	1	1	0.9231	1	1	1	0.9231	1
Q	1	1	∅	0	1	1	1	0.8833
R	1	1	1	1	1	1	1	1
S	1	1	∅	0	1	1	1	0.9917
T	1	1	0.6630	1	1	1	1	1
U	1	1	1	1	1	1	1	1
V	1	1	1	1	1	1	1	1
W	1	1	1	1	1	1	1	1
X	1	1	0.96	1	1	1	0.96	1
Y	1	1	1	1	1	1	1	1
Promed.	1	1	0.8112	0.875	1	1	0.9814	0.9743
F-meas.	1		0.8419		1		0.9778	

En la Tabla N° 11 se muestran de igual manera los valores de precisión y recall de la red de señas y de la red conjugada al aplicarles la señal con 1% de ruido en cada técnica de clustering. Los errores siguen persistiendo en la red de señas, en las señas identificadas como críticas y sin existir muchas variaciones; para la red conjugada, los errores persisten, con una ligera ayuda, resultado de agregar los datos obtenidos de la red MPU.

Tabla 11: Aplicación de ruido al 1% sobre redes neuronales

Letras	Red Señas				Red Conjugada			
	Red GR		Red LVQ		Red GR		Red LVQ	
	Precisión	Recall	Precisión	Recall	Precisión	Recall	Precisión	Recall
A	1	1	1	1	1	1	1	1
B	1	1	1	1	1	1	1	1
C	1	1	1	1	1	1	1	1
D	1	1	1	1	1	1	1	1
E	1	1	1	1	1	1	1	1
F	1	1	1	1	1	1	1	1
G	1	1	∅	0	1	1	1	0.9667
H	1	1	1	1	1	1	1	1
I	1	1	1	1	1	1	1	1
K	1	1	1	1	1	1	1	1
L	1	1	0.3468	1	1	1	1	1
M	1	1	1	1	1	1	1	1
N	1	1	1	1	1	1	1	1
O	1	1	0.5063	1	1	1	1	1
P	1	1	1	1	1	1	0.9302	1
Q	1	1	∅	0	1	1	1	0.9167
R	1	1	1	1	1	1	1	1
S	1	1	∅	0	1	1	1	0.9750
T	1	1	0.9677	1	1	1	0.9677	1
U	1	1	1	1	1	1	1	1
V	1	1	1	1	1	1	1	1
W	1	1	1	1	1	1	1	1
X	1	1	0.9677	1	1	1	0.9677	1
Y	1	1	1	1	1	1	1	1
Promed.	1	1	0.8245	0.875	1	1	0.9944	0.9941
F1	1		0.8490		1		0.9942	

De manera análoga la Tabla N° 12 expone los valores de precisión y recall de la red de señas y de la red conjugada al aplicarles la señal con 5% de ruido en cada técnica de clustering. A partir de este punto ya se pueden apreciar errores en la red GR, los cuales no pueden ser mejorados con los datos obtenidos de la red MPU según lo obtenido de su red Conjugada; de igual manera para la técnica LVQ, sus redes presentan un aumento en sus errores de clasificación, que obtienen una mejora con los datos MPU (un ejemplo de esto es lo ocurrido con las letras Q y S, que aumentan sus niveles de precisión y recall).

Tabla 12: Aplicación de ruido al 5% sobre redes neuronales

Letras	Red Señas				Red Conjugada			
	Red GR		Red LVQ		Red GR		Red LVQ	
	Precisión	Recall	Precisión	Recall	Precisión	Recall	Precisión	Recall
A	1	1	1	1	1	1	0.9302	1
B	1	1	1	1	1	1	1	1
C	1	1	1	1	1	1	1	1
D	1	1	1	1	1	1	1	1
E	1	1	1	1	1	1	1	1
F	1	1	1	1	1	1	1	1
G	0.8219	1	∅	0	0.8219	1	1	0.925
H	1	1	1	1	1	1	1	1
I	1	1	1	1	1	1	1	1
K	1	1	1	1	1	1	1	1
L	0.9023	1	0.3582	1	0.9023	1	1	1
M	1	1	1	1	1	1	1	1
N	1	1	1	1	1	1	1	1
O	1	1	0.5405	1	1	1	0.96	1
P	1	1	0.9023	1	1	1	0.9023	1
Q	1	0.675	∅	0	1	0.675	1	0.867
R	1	0.825	1	0.825	1	0.825	1	0.825
S	1	1	∅	0	1	1	1	0.8500
T	1	1	0.9449	1	1	1	0.8449	1
U	1	1	1	1	1	1	1	1
V	0.8511	1	0.8511	1	0.8511	1	0.8511	1
W	1	1	1	1	1	1	1	1
X	1	1	0.9302	1	1	1	0.9302	1
Y	1	1	1	1	1	1	1	1
Promed.	0.9823	0.9792	0.8136	0.8677	0.9823	0.9792	0.9758	0.9778
F1	0.9807		0.8398		0.9807		0.9768	

Finalmente, la Tabla N° 13 presenta los valores de precisión y recall de la red de señas y de la red conjugada al aplicarles la señal con 5% de ruido en cada técnica de clustering. En esta tabla se puede apreciar una gran variación en los resultados tanto en las redes con técnica GR como las de técnica LVQ.

Tabla 13: Aplicación de ruido al 10% sobre redes neuronales

Letras	Red Señas				Red Conjugada			
	Red GR		Red LVQ		Red GR		Red LVQ	
	Precisión	Recall	Precisión	Recall	Precisión	Recall	Precisión	Recall
A	1	1	0.8108	1	1	1	0.8108	1
B	0.9756	1	0.9756	1	0.9756	1	0.9756	1
C	1	0.767	1	0.7667	1	0.767	1	0.7667
D	1	1	1	0.6667	1	1	1	0.6667
E	1	0.95	1	0.9083	1	0.95	1	0.9083
F	0.8069	0.975	0.8069	0.975	0.8069	0.975	0.8069	0.975
G	0.7073	0.725	∅	0	0.7073	0.725	1	0.733
H	1	1	1	1	1	1	1	1
I	1	0.9667	0.9667	0.9667	1	0.9667	1	0.9667
K	0.5911	1	0.5911	1	0.5911	1	0.5797	1
L	0.5217	1	0.3896	1	0.5217	1	1	1
M	0.8392	1	0.8929	0.833	0.8392	1	0.8929	0.833
N	1	0.975	1	0.975	1	0.975	1	0.975
O	0.5511	0.808	0.5438	0.983	0.5511	0.808	1	0.983
P	1	1	0.8054	1	0.9375	1	0.7101	1
Q	1	0.067	0.16	0.033	∅	0	1	0.833
R	1	0.45	1	0.45	1	0.45	0.9153	0.45
S	1	0.35	0	0	∅	0	0.9898	0.8167
T	0.7987	0.9917	0.8317	0.7	0.623	0.9917	0.7113	0.8417
U	0.9677	1	0.9677	1	0.9677	1	0.9677	1
V	0.5943	0.525	0.5943	0.525	0.5943	0.525	0.5943	0.525
W	1	1	1	1	1	1	1	1
X	1	0.7917	0.7055	0.8583	1	0.7917	0.7055	0.8583
Y	1	1	1	1	1	1	1	1
Promed.	0.8897	0.8476	0.7518	0.7767	0.7965	0.8302	0.9025	0.8805
F1	0.8681		0.7640		0.8130		0.8914	

CONCLUSIONES

En este capítulo se presentan las conclusiones respecto a cada objetivo específico del presente trabajo:

1. Una Red Neuronal de Regresión Generalizada (GRNN), variación de una red de base radial, requiere para su implementación computacional del comando `newgrnn` descrito en la Tabla 3, con parámetros como el vector de prueba, el vector de clases (definido por el usuario) y el parámetro “SPREAD”, el cual no requerimos del cambio de su valor estándar debido a que todas las categorías tienen la misma importancia y necesitan las mismas salidas para el caso particular de la investigación, como se puede apreciar, con el valor estándar no se tuvieron complicaciones. Al implementar distintas redes GR, se pudo comprobar de manera experimental que el cálculo de los pesos neuronales se da por la misma red en función de los vectores de prueba y de categoría previamente impuesto, debido a que las redes presentaban los mismos pesos neuronales, obedeciendo a una regla matemática.
2. Una Red Neuronal de Aprendizaje por Cuantificación Vectorial (LVQNN) puede ser implementada computacionalmente mediante el comando “`lvqnet`” (anteriormente “`newlvq`”, estos comandos tienen diferencias en los vectores de entrada) el cual es descrito junto con sus parámetros de entrada en la Tabla 4. Se debe tener en consideración el valor del parámetro LR, ya que influye en el aprendizaje de la propia red neuronal: mientras más alto sea su valor el aprendizaje será más rápido, pero menos seguro; en el caso contrario, un valor LR más bajo hace un proceso de aprendizaje más lento, pero más fino y seguro; en el caso de la presente investigación, con 24 categorías, el valor LR se mantuvo 10 veces por debajo del valor estándar.
3. La evaluación de los algoritmos implementados, para la conformación de una red neuronal, mediante sus respectivas tablas de contingencia brinda una idea clara y gráfica de la actuación de las distintas redes neuronales al ser sometidas tanto a los datos pre procesados como también al ser sometidas a los datos afectados por ruido. El beneficio de armar estas tablas de contingencia con el comando “`plotconfusion`” perteneciente al Toolbox de MatLab, descrito en la expresión (8), es que nos presenta los valores de precisión y recall dentro de la misma tabla en formas de porcentaje, colocándose los valores de precisión por categoría al lado derecho de la tabla y los

valores de recall en la parte inferior también para cada categoría, pudiendo de esta manera verificar los datos que han sido mal clasificados y en que categoría es que han sido clasificados de manera errónea.

4. La precisión en una red neuronal es la relación entre la cantidad de elementos correctamente clasificados sobre la cantidad total de elementos clasificados en una categoría específica. La evaluación de la precisión característica de los algoritmos implementados, para la conformación de una red neuronal, indica que este parámetro puede variar de 0 a 1 y se puede interpretar como la capacidad de clasificar correctamente elementos en una categoría específica evitando la acumulación de elementos pertenecientes a otras categorías de manera errónea, en caso de obtener un valor no determinado significa que la red neuronal no considera la existencia de elementos que puedan pertenecer a una categoría específica.

Con la aplicación de ruido, los datos se comienzan a acumular erróneamente en distintas categorías, la red GR no se inmuta hasta la aplicación del 5% de ruido (Tabla 12), en donde su precisión disminuye al 0.9823 con errores en 3 categorías que no pueden ser corregidas con los datos MPU, manteniéndose este valor en su red conjugada; con la aplicación del 10 % de ruido (Tabla 13) los elementos se distribuyen a lo largo de 11 categorías distintas, siendo la categoría 11 (letra L) la que acumula una mayor cantidad de elementos erróneos, obteniendo un valor de 0.5217, que no obtiene mejoras con la aplicación de los datos de la red MPU, obteniendo como resultado final de la Red conjugada algunos valores de clasificación correcta nulos y con ello la no existencia de precisión en algunas categorías, disminuyendo de esta manera el valor promedio de precisión de 0.8897 en la Red de señas a 0.7965 en la Red conjugada.

Se pudo comprobar que las redes LVQ son más susceptibles al ruido debido a la cantidad de errores arrojados por la red desde la aplicación del 0.1% de ruido (Tabla 10), con mejoras sustanciales tras la adición de los datos clasificados por la red MPU en la red conjugada GR que se le adicionó variando de 0.8112 en la red de señas a 0.9814 en la Red conjugada; con la aplicación del 5% de ruido (Tabla 12) los errores se vuelven más notables en la Red de señas con una precisión promedio de 0.8136, siendo sustancialmente corregidos en la Red conjugada con valores de precisión muy cercanos a la unidad en todas las categorías y una precisión promedio de 0.9758, subido el ruido al 10% (Tabla 13) se obtiene un mayor aumento de fallas en la

clasificación en la Red de señas, con un promedio de 0.7518, pero con una mejora significativa en la Red conjugada con el aumento de los resultados de la Red MPU, obteniendo un aumento de la precisión promedio a 0.9025.

5. El parámetro recall en una red neuronal es la relación entre los elementos correctamente clasificados sobre la cantidad total de elementos que deberían existir en una categoría específica. La evaluación del recall característico de los algoritmos implementados, para la conformación de una red neuronal, evidencia que este parámetro puede tomar valores desde 0 hasta 1, pudiéndose interpretar como la suficiencia con la cual una red neuronal clasifica correctamente los elementos de una categoría específica sin dispersarlos en otras categorías de manera equivocada, un valor no determinado implica la no existencia de elementos pertenecientes a una categoría en específico; este parámetro es inversamente proporcional a la precisión. Según lo expuesto en las Tablas 10 y 11, se puede observar el correcto funcionamiento de la red GR ante la presencia de ruido hasta un nivel del 5% (Tabla 12), en donde los errores de clasificación se presentan en las categorías 16 y 17 (letras Q y R) y podemos entender que sus elementos han sido dispersados a lo largo de las otras 3 categorías con baja precisión, el valor promedio de recall se mantiene en 0.9792 tanto para la Red de señas como para la Red conjugada; con la aplicación del 10% de ruido (Tabla 13) los datos dispersos pertenecen a 13 categorías distintas que con la aplicación de los datos de la Red MPU en la Red conjugada no obtienen mejoras, en concreto, hay 2 categorías donde se desciende a un valor nulo, las categorías 16 y 18 (letras Q y S), descendiendo el valor promedio de recall de 0.8476 en la Red de señas a 0.8302 en la Red conjugada.

En la red LVQ al aplicar un ruido de 0.1 y 1%, los datos dispersos pertenecen únicamente a las categorías 7, 16 y 18 (letras G, Q y S respectivamente), el valor promedio de recall se ve mejorado ante la adición de los datos MPU en la Red conjugada, mejorando de 0.875 a 0.9743 (Tabla 10) y de 0.875 a 0.9941 (Tabla 11) respectivamente en cada caso; ante la aplicación del 5 y 10% de ruido, aumentan las categorías que sufren la dispersión de sus elementos, manteniéndose en este grupo las categorías 7, 16 y 18 mencionadas anteriormente y sumándose la categoría 17 en el primer caso y una mayor cantidad en el último, los valores de recall promedio obtienen unas mejoras con la aplicación de los datos MPU en la Red conjugada GR que se le adicionó, en el caso del ruido al 5% se observa una mejora de este valor de

0.8677 a 0.9778 (Tabla 12) y en el caso del ruido al 10% una mejora de 0.7767 a 0.8805 (Tabla 13).

De los valores promedio de las redes neuronales con datos pre-procesados (Tabla 9), se observó que la magnitud-F, en todas las redes neuronales GR fue igual a la unidad, significando son más precisas en su clasificación al realizar el cálculo numérico, al punto de no necesitar los datos de su red neuronal MPU para discriminar entre las señales consideradas como críticas; por el contrario, las redes LVQ si pueden notar (con un mayor alcance) similitudes entre los patrones, teniendo así un mayor nivel de confusión en las señas consideradas como críticas, lo cual se logró corregir con los datos obtenidos de la Red MPU; esto, sumado a lo obtenido con la adición de ruido y los bajos niveles de tolerancia aceptados por ambas redes (1% para las redes GR y 0.1% para las LVQ), evidencia la necesidad y la importancia del preprocesamiento de datos para una mayor precisión en la clasificación de los vectores de datos, así como la importancia de una acertada y precisa toma de datos.

RECOMENDACIONES

En esta sección se realizarán algunas recomendaciones para futuras investigaciones que deben ser consideradas para mejoras sustanciales en este tipo de implementación, de manera que, en un futuro, estas tecnologías puedan ser de mayor provecho.

1. Ante la cantidad de errores presentadas por las redes, se pudo verificar que algunas de ellas estuvieron dentro de lo previsto por las señas críticas que se mencionaron, como un ejemplo, se obtuvieron errores entre las letras D y X al aplicar el 10% de ruido, un error que no pudo ser corregido ya que ambas tienen una orientación Y+, viéndolas semejantes desde los niveles de voltaje y la orientación, este error era inevitable. Teniendo entonces en cuenta que, ante el aumento de señas a clasificar aumenta la posibilidad de encontrar señas similares que impliquen otorgar más “detalles” a la red neuronal para que esta pueda realizar una mejor clasificación, para ello se recomienda la implementación de otros sensores que puedan ayudar a la identificación integral del comportamiento de la mano, como contactos eléctricos para saber si hay dedos que se están tocando o si el dedo se encuentra en una zona específica. Se debe
2. Se recomienda también la aplicación de sensores flex más largos, teniendo en consideración los dedos a los que deben ir colocados y/o las señas a clasificar, de manera que se cubra la totalidad del largo del dedo para una toma de datos más precisa, esto se relaciona directamente con el tipo de guante que se utiliza, que de preferencia debe ser más ceñido a la mano del usuario.
3. Ante el aumento de señas a clasificar, se sugiere un preprocesamiento más exhaustivo de los datos, teniendo en cuenta que para una red GR se pueden tolerar variaciones de hasta un 1% en sus señales de entrada, y para redes LVQ se aconseja un entrenamiento más “fino” en consecuencia del aumento de sensores para que la red pueda discriminar con más exactitud, colocando cada elemento en su respectiva categoría en su proceso de aprendizaje. No se excluye la posibilidad de mezclar ambas redes, tal y como se hizo en la sección 4.4, de manera que se pueda tener una red mixta que aproveche las características propias de cada red neuronal.

REFERENCIAS BIBLIOGRÁFICAS

- Chinchor, N. (Junio de 1992). MUC-4 Evaluation Metrics. *Association for Computational Linguistics*, 22-29. doi:10.3115/1072064.1072067
- CONADIS, M. O. (Septiembre de 2016). *Situación de las Personas con Discapacidad Auditiva*. Recuperado el 30 de julio de 2020, de Tiempo Real: <https://www.conadisperu.gob.pe/observatorio/estadisticas/informe-tematico-n-5-situacion-de-las-personas-con-discapacidad-auditiva-en-el-peru/>
- Csoka, F. P. (2017). Key Frame extraction from video sequences containing ASL Sign with Concealed Transmission Errors. *Internacional Convergence on Control, Decision and Information Technologies (CoDIT)*. Barcelona, España: IEEE. doi:10.1109/CoDIT.2017.8102592
- del Brio, B. M. (2002). *Redes Neuronales y Sistemas Difusos* (2 ed.). Madrid, España: RA-MA. Recuperado el Julio de 2020
- DIGEBE. (2015). *Lengua de Señas Peruana* (2 ed.). Lima, Perú: Finishing S.A.C. Recuperado el 22 de Julio de 2020, de <http://repositorio.minedu.gob.pe/handle/MINEDU/5545>
- El Peruano. (21 de Mayo de 2010). *El Peruano*, Web. (Editora Peru) Recuperado el 27 de julio de 2020, de Diario El Peruano: <https://busquedas.elperuano.pe/normaslegales/ley-que-otorga-reconocimiento-oficial-a-la-lengua-de-senas-p-ley-n-29535-496850-3/>
- Elektorstore. (s.f.). *Elektorstore*, Web. Recuperado el 26 de Septiembre de 2019, de Elektorstore: <https://www.elektor.com/arduino-nano>
- Enseñas Perú. (2015). *ONG ENSEÑAS PERU*. Obtenido de <https://www.ensenasperu.org/programa3.php>
- Fernández Suárez, K. J. (2017). *Diseño y construcción de un prototipo de sistema electrónico para conversión*. Tesis para Título Profesional de Ingeniero Electrónico, Universidad Nacional Pedro Ruiz Gallo, Facultad de Ciencias Físicas y Matemáticas, Lambayeque, Peru. Recuperado el 18 de julio de 2020, de <http://repositorio.unprg.edu.pe/handle/UNPRG/1158>

- Fundación Telefonica. (20 de 4 de 2017). *FundacionTelefonica*. Recuperado el 4 de abril de 2020, de <https://educared.fundaciontelefonica.com.pe/conoce-la-lenguade-senas-peruana/>
- Han, J. (2012). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.
- Haykin, S. (2008). *Neural Network and Learning Machines* (3 ed.). New Jersey, New Jersey: Pearson. Recuperado el julio de 2020
- Hebb, D. O. (1949). *The organization of behavior; a neuropsychological theory*. Wiley.
- Hernandez, R. (2014). *Metodologia de la Investigacion Cientifica* (6 ed.). (S. INTERAMERICANA EDITORES, Ed.) México D.F., Mexico: McGrawHill. Recuperado el 15 de julio de 2020, de <http://observatorio.epacartagena.gov.co/wp-content/uploads/2017/08/metodologia-de-la-investigacion-sexta-edicion.compressed.pdf>
- Huamaní, P. (Abril de 2008). Cuaderno de Ingeniería: Simulación de redes neuronales con MatLab. *Taller de impresiones de la Universidad Ricardo Palma*.
- Kai-Yin, F. N.-T. (29 de octubre de 2015). A Real-Time ASL Recognition System Using Leap Motion Sensors. *International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*. Xi'an, China: IEEE. doi:10.1109/CyberC.2015.81
- Keerthi, S. W. (2016). Software based sign language converter. *International Conference on Communication and Signal Processing*. Melmaruvathur, India: IEEE. doi:10.1109/ICCSP.2016.7754472
- Kim, P. (2017). *MATLAB Deep Learning with Machine Learning, Neural Networks and Artificial Intelligence*. Seoul, Soul-t'ukpyolsi, Republica de Corea: Apress. doi:DOI 10.1007/978-1-4842-2845-6
- Kohonen, T. (Septiembre de 1990). The Self-Organizing Map. *Proceedings of the IEEE*, 78(9), 1464 - 1480. doi:10.1109/5.58325
- Lin Monica, V. R. (2014). *Sign Language Glove*. (Cornell University) Recuperado el 15 de Julio de 2020, de Cornell University:

https://people.ece.cornell.edu/land/courses/ece4760/FinalProjects/f2014/rdv28_mjl256/webpage/

- López-Noriega, J. E.-V.-C. (8 de diciembre de 2014). Glove-Based Sign Language Recognition Solution to Assist Communication for Deaf Users. *11th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*. doi:10.1109/ICEEE.2014.6978268
- MathWorks. (2020). *MathWorks*. (M. Works.Inc, Productor) Recuperado el 03 de 08 de 2020, de <https://la.mathworks.com/products/deep-learning.html>
- McCulloch, W. S. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5, 115–133. doi:10.1007/BF02478259
- OMS. (15 de Marzo de 2019). *Organizacion Mundial de la Salud*. Recuperado el 15 de julio de 2020, de WHO: <https://www.who.int/es/news-room/fact-sheets/detail/deafness-and-hearing-loss>
- Otazú, J. C. (26 de 10 de 26). *PeruAccesible*. Recuperado el 4 de agosto de 2020, de LA LENGUA DE SEÑAS ES UN DERECHO: <http://blog.pucp.edu.pe/blog/peruaccesible/2011/10/26/la-lengua-de-senas-es-un-derecho/>
- Praatek, S. S. (2018). Dynamic Tool for American Sign Language Finger Spelling Interpreter. *International Conference on Advances in Computing, Communication Control and Networking*. Greater Noida (UP), India, India: IEEE. doi:10.1109/ICACCCN.2018.8748859
- Sausa, M. (3 de 12 de 2015). *Peru21*. (E. Vargas, Editor, & Grupo El Comercio) Recuperado el 29 de julio de 2020, de Redaccion Peru 21: [https://peru21.pe/lima/peru-hay-532-000-personas-sordas-23-interpretes-video-199711-noticia/#:~:text=En%20el%20Per%C3%BA%20solo%20existen,Int%C3%A9rpretes%20del%20Per%C3%BA%20\(Asisep\).&text=%22En%20Latinoam%C3%A9rica%20ya%20hay%20nueve,estamos%20muy%20atr](https://peru21.pe/lima/peru-hay-532-000-personas-sordas-23-interpretes-video-199711-noticia/#:~:text=En%20el%20Per%C3%BA%20solo%20existen,Int%C3%A9rpretes%20del%20Per%C3%BA%20(Asisep).&text=%22En%20Latinoam%C3%A9rica%20ya%20hay%20nueve,estamos%20muy%20atr)
- Specht, D. F. (noviembre de 1991). A General Regression Neural Network. *IEEE TRANSACTIONS ON NEURAL NETWORKS*, 2(6), 568-576. doi:10.1109/72.97934

- Swets, J. A. (1969). Effectiveness of information retrieval methods. *American Documentation*, 20(1), 72-89. Recuperado el 7 de Septiembre de 2020, de <https://apps.dtic.mil/sti/pdfs/AD0656340.pdf>
- Thalange, A. D. (2015). Effect of thinning extent on ASL number recognition using open-finger distance feature measurement technique. *International Conference on Signal Processing and Communication Engineering Systems*. Guntur, India: IEEE. doi:10.1109/SPACES.2015.7058299
- Vasileios, H. a. (1993). Towards the automatic identification of adjectival scales: Clustering adjectives according to meaning. *Proceedings of the 31st Annual Meeting on Association for Computational Linguistics*, 172–182. doi:10.3115/981574.981597
- Walczak, S. (2003). Artificial Neural Networks. En R. Meyers (Ed.), *Encyclopedia of Physical Science and Technology* (3 ed., págs. 631-645). Academic Press. doi:10.1016/B0-12-227410-5/00837-1
- Walde, S. S. (2003). *Experiments on the Automatic Induction of German Semantic Verb Classes*.
- Yue_Gang Song, Y.-L. Z.-J. (2018). Neural Networks for stock price prediction. *Journal of Difference Equations and Applications*, 1-14.