

Breve reseña sobre el uso de Machine Learning para el procesamiento digital de señales en la actualidad

Brief review on the use of Machine Learning for digital signal processing today.

Christian Llanes López^{1,*} Cynthia de la Caridad Cruz García

¹Facultad de Ingeniería Automática y Biomédica, Universidad Tecnológica de la Habana “José Antonio Hecheverría” (CUJAE), calle 114 No 11901 entre 119 y 127, Marianao, La Habana, Cuba.

*Correspondencia: abchristill@automatica.cujae.edu.cu

Este documento posee una [licencia Creative Commons Reconocimiento/No Comercial 4.0 Internacional](#)



Resumen

La tecnología médica ha evolucionado a pasos agigantados en las últimas décadas; uno de los mayores exponentes de dicho desarrollo es la incorporación de las técnicas de Machine Learning al procesamiento digital de señales. Este trabajo está orientado a mostrar la situación actual del uso de esta técnica de análisis de datos, enfocándose en sus principales características y el procedimiento más reciente de aplicación de la misma, a través de una revisión bibliográfica. Además, se enumeran varias de sus aplicaciones en este campo.

Palabras clave: Biomédica, Machine Learning, Redes Neuronales, Procesamiento de Señales, Señales Medicas

Abstract

Medical technology has evolved by leaps and bounds in recent decades; one of the greatest exponents of this development is the incorporation of Machine Learning techniques into digital signal processing. This work is oriented to show the current situation of the use of this data analysis technique, focusing on its main characteristics and the most recent procedure of its application, through a bibliographic review. In addition, several of its applications in this field are listed.

Keywords: Biomedical, Machine Learning, Neural Networks, Signal Processing, Medical Signal

1. Introducción

El análisis de señales es una tecnología imprescindible en el desarrollo actual de múltiples productos, muchos de los cuales tiene finalidades médicas. En la actualidad la minería de datos representa un recurso importante en el procesamiento de datos, con ello se logra identificar patrones y relaciones ocultas en los datos, los cuales permiten la creación de modelos que ayudan a un mejor entendimiento y diagnóstico de ciertas patologías como la epilepsia.

El empleo de Machine Learning con estos fines, es sin duda un gran beneficio en cuanto a duración del proceso, eficiencia y exactitud de los resultados. Actualmente se pueden analizar distintos tipos de señales entre las que se encuentran: señales cerebrales, electromiográficas de superficie, mioeléctricas y generadas a partir de comandos de voz, etc. Esta tecnología se basa sobre todo en la creación y entrenamiento de Redes Neuronales Artificiales; este sistema es capaz de aprender con el tiempo. El sistema será capaz de mejorar de manera progresiva el desempeño de sus tareas y asignaciones mediante las repeticiones de estas, siendo empleados para la detección de patrones en las señales.

Una red neuronal artificial (ANN en Inglés ó RNA en español) consiste en unidades de procesamiento interconectadas de manera densa, llamadas neuronas, por tener un comportamiento similar al de las neuronas biológicas. Las Unidades de procesamiento reciben, procesan y transmiten señales, tal como las neuronas biológicas.

Los principales acontecimientos históricos relacionados con las redes neuronales son:

- Walter Pitts y Warren McCulloch intentaron explicar en 1943 el funcionamiento del cerebro humano, por medio de una red de células.
- En 1949, el fisiólogo Donald Hebb expuso en su libro *The Organization of Behavior* la conocida regla de aprendizaje. Su trabajo tenía que ver con la conductividad de la sinapsis, es decir, con las conexiones entre neuronas.
- En 1951, Minsky y Edmons montaron la primera máquina de redes neuronales, compuesta básicamente de 300 tubos de vacío y un piloto automático de un bombardero B-24 (en desuso). Llamaron a su creación “Sharc”, se trataba nada menos que de una red de 40 neuronas artificiales que imitaban el cerebro de una rata.
- En 1957, Frank Rosenblatt presentó al Perceptrón, una red neuronal con aprendizaje supervisado cuya regla de aprendizaje era una regla planteada por Hebb.
- En 1969 Marvin Minsky y Seymour Paper escribieron un libro llamado *Perceptrons*, en donde definían a estos como caminos sin salida.
- En los años 60 se propusieron otros 2 modelos también supervisados, basados en el Perceptrón de Rosenblatt denominados Adaline y Madaline. En estos, la adaptación de los pesos se realiza teniendo en cuenta el error, calculando como la diferencia entre la salida deseada y la dada por la red, al igual que en el Perceptrón. Sin embargo, la regla de aprendizaje empleada es distinta.
- En los 70 las redes neuronales artificiales surge con la técnica de aprendizaje de propagación hacia atrás o Backpropagation.
- En 1977, James Anderson desarrolló un modelo lineal, llamado asociador lineal, que consistía en unos elementos integradores lineales (neuronas) que sumaban sus entradas.
- En 1982 John Hopfield presentó un trabajo sobre redes neuronales en la Academia Nacional de las Ciencias, en el cual describe con claridad y rigor matemático una red a la que ha dado su nombre, que viene a ser una variación del Asociador Lineal. En este mismo año la compañía Fujitsu comenzó el desarrollo de computadores pensantes para aplicaciones relacionadas con la Robótica.
- En 1985, el Instituto Americano de Física comenzó la “Anual Networks for Computing”.
- En 1987 la IEEE celebró su primera conferencia internacional sobre redes neuronales. En este mismo año se formó la International Neural Network Society (INNS) bajo la iniciativa y dirección de Grossberg en USA, Kohonen en Finlandia y Amari en Japón.[1]

Un concepto que conforma la base del trabajo con redes neuronales es el peso. El peso es una combinación de parámetros o coeficientes, que combinados con los datos de entrada ofrecen un determinado resultado.

Por ejemplo, imagina que estás escuchando una canción en tu teléfono. La canción es una señal analógica, lo que significa que es una onda continua de sonido. El altavoz de su teléfono convierte la señal analógica en una señal digital, que es una serie de números que representan la amplitud y frecuencia de la onda de sonido. Luego, la señal digital es procesada por el amplificador y el altavoz del teléfono, que la vuelve a convertir en una señal analógica que puede escucharse en los oídos.

El procesamiento de señales digitales se puede utilizar para mejorar la calidad del sonido, eliminar el ruido y agregar efectos a la música. Por ejemplo, se puede usar un procesador de señal digital para reducir el ruido de fondo en una grabación o para agregar eco o reverberación a una canción.

En la actualidad se clasifican a partir de 3 de sus características primordiales:

Según su Topología:

- Red Feedforward y Backforward: que son dos tipos de redes de propagación hacia adelante o acíclicas en las que las señales van desde la capa de entrada hacia la de salida, sin existir ciclos ni conexiones, entre neuronas de la misma capa. La primera consiste en calcular la salida de nuestra red con los valores actuales de los pesos. Y la segunda consiste en calcular cuánto influye un cambio en un determinado peso con respecto al error total y actualizarlo teniendo esta relación en cuenta.

En función de su estructura pueden ser de 3 clases:

- Monocapa: Es la red neuronal más sencilla ya que tiene una capa de neuronas que proyectan las entradas a una capa neuronal de salida donde se realizan los cálculos. (Ejemplo: Perceptrón o el Adeline)
- Multicapa: Esta es conformada por múltiples capas, de tal forma que tienen la capacidad para resolver problemas que no son linealmente separables. (Ejemplo: Perceptrón Multicapa)
- Red Recurrente: Que presenta al menos un ciclo cerrado de activación neuronal (Ejemplo: Elman y Hopfield)

Según su Algoritmo de Aprendizaje:

- Aprendizaje Supervisado: Se necesita información preliminar de entrada previamente clasificado o cuya respuesta sea conocida por ejemplo: Perceptrón simple, Backpropagation, Perceptrón multicapa, Memoria Asociativa Bidireccional.
- Aprendizaje Auto organizado: No necesitan un conjunto de datos previamente organizados. Por ejemplo: Memorias Asociativas, Hopfield, La máquina de Boltzman y la máquina de Cauchy, redes de aprendizaje competitivo, Kohonen y redes de resonancia adaptativa (ART).
- Redes Híbridas: Es una combinación de las 2 anteriores en donde se emplea una función de mejora para facilitar la convergencia: Por ejemplo las redes de base radial.
- Aprendizaje Reforzado: Están ubicadas entre el supervisado y el auto organizado [2]

2. Materiales y Métodos

Para este trabajo se realizó una búsqueda bibliográfica en diferentes bases de datos como la biblioteca digital ACM, PubMed, Scielo, Google Scholar etc. y se realizó entrevistas y consultas a profesores y especialistas con gran experiencia en el tema. Se implementaron varios términos de búsqueda como “aprendizaje automático”, “procesamiento digital de señales” y “implementación de Machine Learning” y los resultados se filtraron a artículos publicados en los últimos 10 años en español e inglés.

Como materiales se implementaron mayormente dispositivos electrónicos como computadoras y dispositivos móviles y nos apoyamos también de bibliografía en la biblioteca de la universidad.

3. Resultados y Discusión

En dicha sección se exponen y se discuten todos los resultados obtenidos en el trabajo que se publica.

A continuación se expone la forma de presentar una figura en el manuscrito (válida para cualquier sección del mismo). El aprendizaje automático se trata fundamentalmente de extraer el valor de grandes conjuntos de datos. A menudo, la motivación es producir un algoritmo que pueda imitar o mejorar el desempeño humano. Básicamente consiste en algoritmos que permiten predecir de qué clase es un elemento conociendo las características de éste. Por ejemplo, si se quisiese clasificar frutas en una imagen, sus características podrían ser el color, diámetro, área, centro de masa, etc. Una vez determinada las características hay que predecir el tipo de fruta que se está estudiando, según su parecido con las clases de frutas que puede haber. Las frutas del mismo tipo tendrán características parecidas, pero lo más probable es que no sean iguales. Además, puede haber una característica que sea muy parecida a la de otro tipo, por ejemplo, los plátanos y limones son amarillos, pero no tienen la misma forma. Hay que saber elegir las características y el método de clasificación adecuado según la circunstancia. [4]

Los tipos empleo de Machine Learning son:

- Supervisados: En este tipo se conoce la clase a la que pertenecen los patrones de entrenamiento, las muestras de entrenamiento están etiquetadas. Aprenden funciones que relacionan los datos de entrada y salida.

Ejemplos de supervisados: Análisis discriminante (LDA, QDA y RDA), SVM, Naive Bayes, KNN, Decission Trees, LR, Nearest Archetype, Gradiente descendente, Riemannian manifold.

- No supervisados: Las muestras de entrenamiento no están etiquetadas. El objetivo es aumentar el conocimiento estructural de los datos (clustering, reducción de dimensionalidad o aprendizaje topológico). Estos clasificadores tienden a ser menos precisos que los supervisados.

Ejemplos de no supervisados: K-means, Gaussian EM. Dentro de los no supervisados se encuentra el Aprendizaje de la Representación (Deep Learning) [3]

La metodología teórica general para la clasificación de datos corresponde a: [3]

1. Segmentación de datos Los resultados obtenidos de la señal preprocesada no se consideran una entrada útil en la técnica de reconocimiento de patrones. Así, para extraer las características descriptivas, se requiere la ventana (segmentación) de los datos Preprocesados. Se proponen principalmente dos tipos diferentes de técnicas de creación de ventanas: ventana superpuesta y ventana no superpuesta. En la técnica de ventanas superpuestas, la ventana anterior se superpone a la ventana actual con un incremento atemporal que la longitud de la ventana misma. Una ventana de mayor longitud proporcionaría una alta precisión de clasificación pero retrasaría la decisión del clasificador. La segmentación de los datos usando ventanas ayuda a estimar el movimiento previsto para el clasificador. Ayuda en la toma de decisiones sobre el movimiento previsto mientras se adquieren nuevos datos.

2-Extracción de características: En general, las características de la señal se extraen en forma de dominio de tiempo (TD), dominio de frecuencia (FD) y dominio de frecuencia de tiempo (TFD). En el TD, las características se extraen de las variaciones de la amplitud de la señal con el tiempo según las condiciones musculares. A diferencia de las características temporales, el dominio de la frecuencia usa la densidad del espectro de potencia de las señales para la extracción. Por otro lado, las características combinadas del dominio del tiempo y la frecuencia se utilizan para la extracción de tiempo-frecuencia (ejemplos como la transformada corta de Fourier y las ondículas).

3- Reducción de dimensionalidad (DR): DR es el proceso de eliminar el número de variables arbitrarias bajo consideración mediante la localización de un grupo de variables clave. Los métodos de reducción de dimensionalidad pueden unir esta información de manera más efectiva y resolver el problema de la dimensión de las características. La reducción de la dimensionalidad ayuda a ahorrar el costo computacional y reduce el nivel de complejidad del sistema. El análisis discriminante lineal no correlacionado (ULDA), el análisis de componentes principales (PCA) y el enfoque discriminativo de vecindario difuso ortogonal (OFNDA) son técnicas comunes de reducción de dimensionalidad que se utilizan para reducir el espacio de características. [4]

4- Etiquetar vectores de características: Usualmente los programas computacionales trabajan de mejor manera con números, por lo que es conveniente asignar las muestras de cada paciente a valores numéricos. Este proceso también se denomina asignación de clases o etiquetas.

5- Asignar conjuntos de entrenamiento, validación y prueba: Esta etapa requiere la división de la base de datos en 3 conjuntos:

- Entrenamiento: Permite establecer los parámetros para poder entrenar un modelo de clasificación o clasificador. De esta forma, el conjunto de entrenamiento sirve para establecer como un computador puede procesar la información que se le entrega. [5]
- Validación: Se usan muestras distintas a las del entrenamiento y se le asigna a cada muestra una clase en función de los rasgos que comparten con las muestras de entrenamiento.[3] Este conjunto sirve para establecer los hiperparámetros de un clasificador. Los hiperparámetros son parámetros o valores que son propios de la arquitectura de un clasificador. Por ejemplo, la cantidad de neuronas y capas ocultas de una red neuronal artificial.

• Prueba: Se utiliza netamente para determinar el rendimiento del clasificador. Generalmente la cantidad de muestras en el conjunto de prueba debe ser igual a la del conjunto de validación. No debe ser utilizado para el entrenamiento y ajuste del clasificador. La distribución probabilística de estos conjuntos debe ser idealmente igual para los 3, es decir, las proporciones de las clases en los conjuntos debe ser idéntica a las proporciones de los conjuntos con respecto a la base de datos. [3]

6- Elección de un clasificador: Una vez se tienen los conjuntos para entrenar, ajustar y probar los modelos de clasificación, ahora se procede a elegir el tipo de clasificador a utilizar.[5] Algunos ejemplos de técnicas son : Multi-Layer Perceptron (redes neuronales básicas), Redes Recurrentes, Redes Convolucionales, Support Vector Machine y Linear Discriminant Analysis[6]. En el entrenamiento del clasificador, a partir de la información relevante de un conjunto de muestras, se configuran los parámetros internos que regirán el comportamiento del algoritmo.

Con la finalidad de mantener la brevedad de la reseña en todo lo posible, a pesar de lo extenso del tema, se explicará el funcionamiento básico de un solo tipo de clasificador: Decision Trees.

Un árbol de decisión segmenta el espacio de características en zonas que no solapen, usando la división binaria recursiva. Clasifican las observaciones creando una secuencia de cuestiones sí/no. Primero se consideran todas las observaciones de la misma clase, luego se segmenta el espacio identificando los umbrales de determinadas características (nodo de decisión), para poder separar clases. En cada nodo de decisión, hay que preguntarse si una característica es mayor o menor de un umbral, se considerará ese patrón más semejante a un tipo u otro. Cuando se termina de evaluar una característica, se pasa a la siguiente división del árbol, evaluando otra característica, así hasta llegar a una de las hojas del árbol, en la que se determina a cuál de las clases correspondería el patrón.. Una vez determinados los umbrales, para determinar la clase de un nuevo patrón solo hay que recorrer el árbol respondiendo a las cuestiones de sí/no, según las características correspondientes del patrón estudiado. Hay que impedir que el modelo se amolde en exceso a los datos de entrenamiento (overfitting). Dos posibles soluciones son parada temprana o podado del árbol (reducir el número de divisiones).[3]

7- Análisis de rendimiento de clasificador: Luego de entrenar y ajustar un clasificador con los conjuntos previamente mencionados, se procede a evaluar el conjunto de prueba en el modelo de clasificación obteniendo de esta forma etiquetas de predicción para este conjunto. Luego, se comparan las etiquetas de predicción con las etiquetas actuales del conjunto de prueba obteniendo lo que se conoce como matriz de confusión. Calcular la traza de la matriz, es decir, la suma de los elementos de la diagonal, dividida por el total de muestras en el conjunto de prueba entrega la precisión del clasificador. Para los problemas de clasificación binarios también es válido analizar el rendimiento de cada clase por separado, definiendo los conceptos de sensibilidad, como la capacidad para detectar casos que realmente padecen la enfermedad y la especificidad como la capacidad para detectar casos que realmente sean pacientes sanos.

8- Validación cruzada aleatoria: Para validar el modelo de clasificación, se utilizará una técnica conocida como validación cruzada aleatoria (Repeated random sub-sampling validation). Realizando

este procedimiento de forma independiente para las tareas realizadas por los pacientes permite determinar cuáles son las combinaciones óptimas de procesamiento, extracción de características y clasificadores que maximicen la precisión del modelo de clasificación. Por último, para disminuir el costo computacional del problema (tiempo de procesamiento del algoritmo) y su variación en los resultados de clasificación se deberá realizar un proceso de selección de características, para así filtrar las características que no entreguen información útil al modelo.[5]

Aplicaciones

1. Clasificación de señales EMG:

Las señales EMG se caracterizan por muchas interferencias, como ruido de adquisición de la señal, perturbaciones electromagnéticas, inestabilidad de la señal y artefactos de movimiento debido a electrodos y cables. El preprocesamiento es el primer paso de las técnicas de reconocimiento de patrones con respecto al análisis adecuado de la señal y la minimización de las interferencias inherentes. Cabe señalar que el análisis de componentes independientes (ICA) y el patrón espacial común (CSP) se utilizan como preprocesamiento (filtrado) y reducción de dimensionalidad (después de la extracción de características).[4]

Clasificador de Espectrogramas de Señales EMG con Redes Neuronales Convolucionales –propuesto por Diana del Milagro en su artículo “Análisis de Patrones en Señales Musculares de Extremidad Superior Mediante Técnicas de Machine Learning” –:

A través de la librería TensorFlow™ y su red neuronal convolucional pre entrenada en el reconocimiento de imágenes, se implementa un clasificador de señales EMG. Para ello, se recolectan los espectrogramas de las señales EMG en un mismo directorio, etiquetándolos según el movimiento correspondiente. [1]

Clasificador de Señales EMG con Perceptrón Multicapa -propuesto por la autora Diana del Milagro en su artículo “Análisis de Patrones en Señales Musculares de Extremidad Superior Mediante Técnicas de Machine Learning” de 2019 -: Haciendo uso de la librería Scikit Learn de Python, se implementa un clasificador de señales EMG a partir de la red neuronal de tipo perceptrón multicapa. Los datos recolectados en formato .txt son leídos por una función implementada en Python. Este programa se realiza con la finalidad de convertir los archivos mencionados en arreglos de valores obtenidos. Con este banco de datos, y sus respectivas etiquetas, se realiza el entrenamiento de la red, con la finalidad de que esta sea capaz de diferenciar entre los diferentes tipos de movimientos administrados [1]

Estos modelos empleados en este último artículo no lograron satisfacer las expectativas en dichos artículos. Esto no es debido a que estos no sean adecuados, sino que la data proporcionada no brindaba la suficiente información para ambos casos, destacando la importancia de emplear. Cabe resaltar que, de tener una mayor cantidad de data, y analizarla de tal forma que la información no se vea opacada por data basura, los resultados a obtener, probablemente, serían mucho mejores.

2. Clasificación de psicosis, clasificación de señales EEG -propuesta por el autor de M. J. L. D. LUIS, en su artículo «PROCESAMIENTO DIGITAL DE SEÑALES DE EEG PARA CLASIFICACIÓN DE TRASTORNOS PSICÓTICOS MEDIANTE APRENDIZAJE DE MÁQUINAS,» en 2021.- :

Las señales cerebrales, en comparación con otros tipos de señales, presentan alta resolución espacial y fidelidad de la señal, resistencia al ruido y robustez sustancial durante largos períodos de grabación, lo que genera una mayor precisión.

Para poder realizar una clasificación de los datos, las características de las señales extraídas, en general, deben estar representadas como un vector de valores numéricos, también conocido como vector de características. Estos vectores de características representan a una muestra de la base de datos en su totalidad, por lo que pueden ser utilizados para entrenar modelos de clasificación utilizando herramientas de aprendizaje de máquinas, también conocidas como Machine/Deep Learning. En general, la clasificación de datos se realiza utilizando algoritmos especializados de Machine Learning, tales como las librerías sk-learn del software de programación Python.[5]

3. Deletreador utilizando P300 (potencial equivocado que puede ser registrado mediante una electroencefalografía como una reflexión positiva del voltaje) BCI:

Los datos se recopilaron de los sujetos utilizando un paradigma de ortografía basado en P300. En el experimento, se presenta al usuario en la pantalla de una computadora una matriz de 6 por 6 que incluye caracteres como letras y números. Luego, el usuario se enfoca en un carácter mientras cada fila o columna de la matriz parpadea en un orden aleatorio. Doce destellos forman una ejecución que cubre todas las filas y columnas de la matriz. La tarea del deletreador es identificar el carácter deseado del usuario en función de los datos de EEG recopilados en estos destellos. El conjunto de datos contiene datos de entrenamiento y datos de prueba recopilados de los sujetos. El sistema se entrena por separado para cada usuario usando una frase común con 41 caracteres “EL RÁPIDO ZORRO MARRÓN SALTA SOBRE EL PERRO PEREZOSO 246138 579”. La misma frase también se usa para probar la recopilación de datos con un orden de palabras aleatorio. En este caso se emplea un clasificador SVM entrenado utilizando un conjunto de datos de entrenamiento.[7]

VENTAJAS DE LAS RNA

- Aprendizaje Adaptativo: Las RNA aprenden a realizar tareas a partir de un conjunto de datos dados, en el proceso de aprendizaje, estos datos son representados como las entradas y pesos. Es una de las características más atractivas de redes neuronales. Esto es, aprenden a llevar a cabo ciertas tareas mediante un entrenamiento con ejemplos ilustrativos. Como las redes neuronales pueden aprender a diferenciar patrones mediante ejemplos y entrenamientos, no es necesario elaborar modelos a priori ni necesidad de especificar funciones de distribución de probabilidad.
- Auto-organización: Pueden crear su propia organización o representación de la información recibida. Las redes neuronales auto-organizan la información que reciben durante el aprendizaje y/o la operación:

Mientras que el aprendizaje es la modificación de cada elemento procesal, la auto-organización consiste en la modificación de la red neuronal completa para llevar a cabo un objetivo específico. Cuando las redes neuronales se usan para reconocer ciertas clases de patrones, ellas auto organizan la información usada. Por ejemplo, la red llamada backpropagation, creará su propia representación característica, mediante la cual puede reconocer ciertos patrones. Esta auto organización provoca la generalización: facultad de las redes neuronales de responder apropiadamente cuando se les presentan datos o situaciones a las que no había sido expuesta anteriormente. El sistema puede generalizar la entrada para obtener una respuesta. Esta característica es muy importante cuando se tiene que solucionar problemas en los cuales la información de entrada no es muy clara; además permite que el sistema dé una solución, incluso cuando la información de entrada está especificada de forma incompleta.

- Operación en tiempo real: Las operaciones realizadas pueden ser llevadas a cabo por computadores paralelos, o dispositivos de hardware especiales que aprovechan esta capacidad. Las redes neuronales fueron los primeros métodos computacionales con la capacidad inherente de tolerancia a fallos. Comparados con los sistemas computacionales tradicionales, los cuales pierden su funcionalidad cuando sufren un pequeño error de memoria, en las redes neuronales, si se produce un fallo en un número no muy grande de neuronas y aunque el comportamiento del sistema se ve influenciado, no sufre una caída repentina.
- Tolerancia a fallos parciales: La destrucción parcial de una red, daña el funcionamiento de la misma, pero no la destruye completamente. Esto es debido a la redundancia de la información contenida.
- En el proceso de aprendizaje, los enlaces ponderados de las neuronas se ajustan de manera que se obtengan ciertos resultados específicos. Una red neuronal no necesita un algoritmo para resolver un problema, ya que ella puede generar su propia distribución de pesos en los enlaces mediante el aprendizaje. También existen redes que continúan aprendiendo a lo largo de su vida, después de completado su período de entrenamiento. Hay dos aspectos distintos respecto a la tolerancia a fallos: a) Las redes pueden aprender a reconocer patrones con ruido, distorsionados o incompletos. Esta es una tolerancia a fallos respecto a los datos. b) Las redes pueden seguir realizando su función (con cierta degradación) aunque se destruya parte de la red. La razón por la que las redes neuronales son tolerantes a los fallos es que tienen su información distribuida en las conexiones entre neuronas, existiendo cierto grado de redundancia en este tipo de almacenamiento. La mayoría de los ordenadores algorítmicos y sistemas de recuperación de datos almacenan cada pieza de información en un espacio único, localizado y direccionable. En cambio, las redes neuronales almacenan información no localizada. Por lo tanto, la mayoría de las interconexiones entre los nodos de la red tendrán sus valores en función de los estímulos recibidos, y se generará un patrón de salida que represente la información almacenada.
- Operación en tiempo real: los cómputos neuronales pueden ser llevados a cabo en paralelo; para esto se diseñan y fabrican máquinas con hardware especial para obtener esta capacidad.

- Inclusión flexible en la tecnología vigente: se pueden obtener chips especializados para redes neuronales que mejoran su capacidad en ciertas tareas. Ello permitirá la integración de módulos en los sistemas existentes.
- Las redes neuronales se autoajustan a los elementos procesales: Son dinámicas, pues son capaces de estar constantemente cambiando para adaptarse a las nuevas condiciones. • La función del diseñador es únicamente la obtención de la arquitectura apropiada. No es problema del diseñador el cómo la red aprenderá a discriminar. Sin embargo, sí es necesario que desarrolle un buen algoritmo de aprendizaje que le proporcione a la red la capacidad de discriminar, mediante un entrenamiento con patrones.

DESVENTAJAS DE LAS RNA

Las desventajas pueden ser:

- Complejidad de aprendizaje para grandes tareas, cuantas más cosas se necesita que aprenda la red, más complicado será enseñarle
- Tiempo de aprendizaje elevado. Esto depende de dos factores: primero se incrementa la cantidad de patrones a identificar o clasificar y segundo se requiere mayor flexibilidad o capacidad de adaptación de la red neuronal para reconocer patrones que sean sumamente parecidos, se deberá invertir más tiempo en lograr que la red converja a valores de pesos que representan lo que se quiere enseñar.
- Elevada cantidad de datos para el entrenamiento, cuanto más flexible se requiera que sea la red neuronal, más información tendrá que enseñarle para que realice de forma adecuada la identificación. [10]

4. Conclusiones

La importancia del machine learning en el procesamiento digital de señales se pudo constatar mediante la búsqueda bibliográfica de una amplia gama de ejemplos.

Las redes neuronales también son adecuadas para los sistemas en tiempo real, por las características de su arquitectura en paralelo, permitiendo así ejecutar cálculos a una gran velocidad y a sus tiempos de respuesta.

El uso de Machine Learning orientado al procesado de señales en tiempo real minimiza el proceso de filtrado en comparación con técnicas aplicadas tradicionalmente. Brinda mayor seguridad en los resultados obtenidos y proporciona un alto grado de personalización en los equipos médicos.

Por lo expuesto anteriormente y, debido a su bajo coste computacional, es una técnica que ha convencido a los investigadores para utilizarlos en la clasificación de señales biomédicas.

Referencias

1. Parajuli, N.; Sreenivasan, N; Bifulco, P; Cesarelli, M.; Savino, S.; Niola, V.; Esposito, D.; Hamilton, T.; Naik, G.; Gunawardana, U. & Gargiulo, G., *Real-Time EMG Based Pattern Recognition Control for*

- Hand Prostheses: A Review on Existing Methods, Challenges and Future Implementation.* MDPI, 2019, p. 30.
2. Ramirez, I, *Machine Learning para MI-BCI orientada al procesado de las señales EEG en tiempo real*, 2021. Escuela Técnica Superior de Ingeniería Universidad de Sevilla.
 3. Sánchez, E.; Varela, A. & Campbells, E., *Redes Neuronales Artificiales: Una Revisión del Estado del Arte, Aplicaciones y Tendencias Futuras. Investigación y Desarrollo. TIC*, 2011. p. 18-27
 4. De Luna, M., *Procesamiento Digital de Señales de EEG para clasificación de trastornos psicóticos mediante aprendizaje máquinas*, 2021. Universidad de Chile.
 5. Craik, A.; He, Y.; & Contreras-Vidal, J.; *Deep learning for electroencephalogram (EEG) classification tasks: a review*. Journal of Neural Engineering, 2019.
 6. Devia, C.; Mayol, R.; Parrini, J.; Orellana, G.; Ruiz A.; Maldonado, P.; & Egana, J., *EEG Classification During Scene Free-Viewing for Schizophrenia Detection*, IEEE Transactions of Neural Systems and Rehabilitation Engineering, 2019. p. 1-1.
 7. Galvan, D., *Análisis de Patrones en Señales Musculares de Extremidad Superior Mediante Técnicas de Machine Learning*, 2019. Universidad de Lima.
 8. Li, Y.; Keng Ang, K.; and Guan, C., *Digital Signal Processing and Machine Learning*, 2010. Y. Li et al., p. 305-330.