Aprendizaje incremental para la tarea de reconocimiento de dígitos con Redes Neuronales Artificiales

Fragoso García Sandra
email sfragosog001@alumno.uaemex.mx

González Hernández Luis Ángel
email lgonzalezh014@alumno.uaemex.mx

4 de noviembre de 2022

Resumen

El aprendizaje incremental es un área de la Inteligencia Artificial la cual permite agregar nuevo conocimiento a un modelo (e.g. Redes Neuronales Artificiales) sin la necesidad de entrenar el modelo con toda la información histórica de la tarea en cuestión [3]. En el presente trabajo de investigación se ocupara el modelo de Redes Neuronales Artificiales enfocada en la clasificación de dígitos escritos a mano usando el algoritmo de entrenamiento de backpropagation, con redes Multi capa Perceptron y duplicación de pesos múltiples simulando memoria a corto y largo plazo para mejorar los resultados presentados en [3].

Palabras claves: Aprendizaje Incremental, Redes Multi Capa, Redes Neuronales Artificiales

1. Introducción

La inteligencia artificial (IA) es una área del conocimiento que se enfoca en poder hacer máquinas que tengan un comportamiento y razonamiento humano, para que en un momento dado, se pueda interactuar con ellas sin darse cuenta que se está interactuando con una máquina. Así mismo, también es posible pensar que mucho del desarrollo en el área de inteligencia artificial, es el poder tener mejores herramientas que ayuden a las actividades diarias.

En este sentido, un área de la IA es el llamado Aprendizaje Máquina, donde se estudian algoritmos que permiten aprender de forma automática una tarea. Así, una de las técnicas más conocidas en la actualidad, dentro del área de IA son las Redes Neuronales Artificiales (RNAs), siendo técnicas que realizan procesos matemáticos para poder aprenderse tareas a resolver. Algunas áreas en las que son útiles las RNAs son en el aprendizaje de tareas no lineales, como la predicción de la capacidad de la red 5G, basada en el tráfico diario de este [18] o clasificación, por ejemplo la clasificación de metales y rocas por medio de RNAs y lógica difusa [17].

Las RNAs están formadas por neuronas artificiales que simulan a las biológicas. Así los procesos químicos que suceden en el cerebro, se simulan computacionalmente a través de señales que viajen a través de las neuronas artificiales, de aquí en adelante simplemente se referirá a ellas como ”neuronas”. Las neuronas en una RNA cuentan con una estructura distribuida en paralelo, presentando una buena habilidad de aprendizaje [12].

Dentro del aprendizaje máquina cuando una técnica, por ejemplo RNA, se enfoca en aprender una tarea, se le conoce como *algoritmo lineal sin memoria*, siendo uno de los métodos más empleados desde el inicio de las RNAs [9]. Sin embargo, si es necesario incorporar nueva información del problema, es necesario volver a entrenar todo el modelo, considerando toda la información existente, esto es, la anterior y la nueva que acaba de llegar, es ahí donde nace el concepto de Aprendizaje Incremental, siendo un área enfocada en poder incorporar información del problema en cuestión, sin tener que volver a re-entrenar todo el modelo.

Derivado del aprendizaje incremental se desprende el concepto de memoria dentro de la IA, analizando como un algoritmo de aprendizaje máquina puede olvidar la información que se usó en un entrenamiento previo al entrenar con información más reciente. Si se hace la analogía con los humanos, la memoria es un factor importante para estudiar considerando la perdida de información aprendida, así, este es un problema biológico,

el cual tanto afecta a los humanos como a las máquinas. Por ello, se han elaborado distintos experimentos para poder combatir esta problemática. Uno de estos es el caso de [3], el cual propone el manejo de RNAs con pesos dobles, donde la primer capa de pesos esta enfocada a comportarse como memoria a corto plazo, y la segunda como memoria a largo plazo. Los experimentos mostrados en [3] permiten notar un mejora en tareas de aprendizaje incremental, teniendo menos pérdida de información en comparación de implementaciones anteriores como el algoritmo Learn++ [11, 6].

Así, el presente trabajo de investigación esta enfocado en poder explorar nuevas configuraciones de pesos duplicados para poder extender el trabajo previamente presentado en [3].

2. Planeamiento del Problema

Las Redes Neuronales Artificiales tienen la habilidad de poderse aprender una tarea y poder hacer o resolver tareas de predicción o clasificación básicamente. En este sentido, con algoritmos de aprendizaje como el Backpropagation, permite ajustar los pesos de una RNA para que esta pueda empezar a resolver una tarea particular. No obstante, muchas de las tareas que se resuelven en la vida diaria, van generando mas información con el tiempo, por ejemplo, el comportamiento de un una serie financiera, o bien la predicción del clima en una determinada región. Así, se tiene el aprendizaje incremental, siendo un método poco explorado enfocado en poder aprender nueva información del problema, sin tener que volver a entrenar todo el modelo con la información anterior y la nueva que acaba de llegar, esto es, en los modelos actuales de aprendizaje máquina, si se usa un conjunto de datos para entrenar un modelo en especifico, dicho modelo es funcional para dicho conjunto de datos y la información que ello representa. Sin embargo, si es necesario incorporar nueva información al modelo, es necesario recolectar dicha información nueva, agregarla a la que se tenia anteriormente, y volver a entrenar todo el modelo para que éste pueda incorporar la nueva información.

De esta forma, una red ya entrenada con un primer conjunto de datos d_1 se desea entrenar con un nuevo conjunto de datos del mismo problema d_2, al entrenar la red con d_2 se perderá en conocimiento aprendido por d_1. Si no se desea utilizar un modelo de aprendizaje incremental, se tendría que juntar el conjunto d_1 y d_2 en un solo conjunto y volver a entrenar la RNA para así poder incorporar el nuevo conocimiento (d_2) a la RNA. Si se desea utilizar el método de aprendizaje incremental, se puede entrenar en un primer momento a la red con el conjunto d_1, posteriormente con d_2 teniendo poca perdida de información de d_1. Si más adelante llega mas información del problema (d_3) que se desee incorporar a la base de conocimientos de la RNA, entonces solo habrá que entrenar la RNA con d_3 usando el modelo incremental para tener una pérdida mínima de información de d_1 y d_2.

De esta forma, el presente trabajo tomará como base la investigación de [3], en donde utiliza una configuración de pesos dobles (a una RNA se duplican todos sus pesos), donde a una capa de pesos duplicados se enlaza con una tasa de aprendizaje alta, para simular un rápido aprendizaje, y por ende simular lo que sería memoria a corto plazo. En su contraparte, la segunda capa de pesos duplicados, se enlaza con una tasa de aprendizaje baja para aprender lentamente un problema, simulando la memoria a largo plazo. Es decir, al momento de aprender una tarea nueva, una capa de pesos aprenderá muy rápido la nueva tarea (tasa alta de aprendizaje) y por ende olvidará más rápidamente los datos ya aprendidos anteriormente, por otro lado, la segunda capa de pesos duplicados, aprenderá muy lentamente los nuevos datos que llegan, y por ende olvidando poco la información que anteriormente se aprendió. Considerando ello y también que la RNA trabaja en conjunto con ambas capas de pesos duplicados, se pondera la salida de la RNA para que pueda contemplar la información nueva que se acaba de agregar a la RNA en ambas capas, así como la información que se acaba de olvidar de ambas capas de pesos.

El problema principal del aprendizaje incremental mostrado en [3], es que entre más conjuntos de datos nuevos que lleguen, mas se olvidaran los primeros conjuntos que se aprendieron, lo cual no es tan útil si se contempla que en el futuro de una RNA podrán existir 10 o 20 etapas de entrenamiento incremental con nuevos conjuntos de datos que se vayan recolectando.

Por ello es indispensable poder explorar nuevas configuraciones de RNAs que permitan mejorar los métodos actuales para permitir una menor cantidad de olvido conforme llegue nueva información al modelo. Donde al igual que en trabajos anteriores, el presente trabajo se basará en conceptos de memoria a corto y largo plazo, y en lugar de hacer una copia de los pesos actuales y tener dos tasas de aprendizaje, una rápida para simular la memoria a corto plazo, y otra tasa de aprendizaje para simular la memoria a largo plazo, se explorará por hacer mas copias de los pesos, teniendo más tasas de aprendizaje que operen en cada una de dichas copias.

3. Objetivos

Diseñar una red neuronal artificial para aprendizaje incremental basada en el principio de la memoria a corto y largo plazo, buscando usar más de dos capas de pesos duplicados para el reconocimiento de dígitos, y con una menor pérdida de información de trabajos previos.

3.1. Objetivos Particulares

1. Implementar el algoritmo mostrado en [3] para el reconocimiento de dígitos con aprendizaje a corto y largo plazo con los parámetros que ahí se indican.
2. Obtener el conjunto de datos de Optical Digits, limpiar los datos y prepararlos según lo indicado con [3].
3. Separar el conjunto de entrenamiento y de prueba de acuerdo a lo que se explica en el artículo de Bullinaria y probar el primer código implementado en miras de comprobar los resultados previamente mostrados en [3].
4. Tomar como base el algoritmo implementado, y extenderlo para permitir mas de dos pesos duplicados, aplicando el conjunto de datos previamente mostrado.
5. Comparar ambas implementaciones en busca de una reducción significativa de las tasas de aprendizaje con respecto a trabajos previos en la literatura.

4. Hipótesis

Al tener más de dos capas de pesos duplicados con sus respectivas tasas de aprendizaje, permite tener un menor olvido de la información previa aprendida, en un modelo de aprendizaje incremental con redes neuronales artificiales del tipo MLP, al usar el algoritmo de Backpropagation.

5. Justificación

Las redes neuronales permiten el aprendizaje automático y la resolución de distintos problemas, pero como se comentó anteriormente, las técnicas de aprendizaje máquina, tienen una deficiencia que es al momento de aumentar los nuevos bloques de datos que llegan para aprender, se obtiene un deterioro en el rendimiento de aprendizaje de información y olvido de la información anterior [3].

Los resultados que se han obtenido no han funcionado a la perfección, la memoria a corto plazo olvida poco pero va olvidando, y lo ideal sería que no olvidara. Biológicamente, los humanos pueden aprender nuevas tareas, o información nueva de un problema, y no olvida de forma significativa lo que anteriormente aprendió, no obstante, eso no pasa actualmente con las RNA y en general con cualquier algoritmo de aprendizaje máquina. En otro sentido, los humanos ya tienen cierta configuración en el cerebro que les permite aprender como se hace actualmente, y se puede afirmar que por el momento no hay ningún procedimiento (ya sea quirúrgico o no) que permita modificar la estructura del cerebro para aprender mas y olvidar menos.

No obstante, computacionalmente nada puede impedir que se experimente con más configuraciones y llegar al punto en donde toda la información que ingrese a un modelo (por ejemplo RNAs) se acumule, y si no hay problema de almacenamiento, que ésta se siga acumulando y que no olvide, esto podría ser bueno en diversas situaciones.

Desde un punto de vista computacional, si llega nueva información y no se ocupa aprendizaje incremental, esto implicará volver a entrenar todo el sistema con la información anterior y la actual (por ejemplo, d_1 y d_2) y considerando que una de las desventajas que tienen la RNAs es que el entrenamiento es un cuello de botella, siendo este donde se lleva la mayor parte de cómputo y por consiguiente de energía. Lo mencionado anteriormente implica que volver a entrenar con toda la información acumulada, gastará más energía y tiempo, que si solo se entrena con la nueva información que llega al modelo. En su contraparte, existe una gran variedad de herramientas, las cuales permiten codificar una red neuronal artificial con librerías ya preexistentes, para esta investigación solo se expondrán 2 empresas, siendo estas las más importantes: Microsoft y Google. La primera cuenta con la plataforma de Azure que renta una maquina virtual donde se puede programar en Python, y la segunda, cuenta con Google Colab que de igual forma brinda una máquina virtual para realizar experimentos

de Maching Learning, la única diferencia contra Azure es que dicha herramienta es gratuita, y una similitud que tienen es que en ambas herramientas se puede programar en el mismo lenguaje.

Como se puede observar, las herramientas mencionadas permiten la programación en Pyhton, y esto se debe a que dicho lenguaje es una herramienta de software libre que no requiere licencia, es relativamente fácil poder depurar un código y permite acelerar más el desarrollo de aplicaciones, a diferencia de otros lenguajes más estructurados como C o Java, además tiene más librerías para el desarrollo de Maching Learning, por ejemplo, TensorFlow, Numpi, entre otras.

TensorFlow es una librería de Python que permite construir y entrenar redes neuronales para detectar patrones y razonamientos usados por los humanos, en la presente investigación se usará dado a que favorece la creación de una RNA, permite la elaboración de cualquier tipo de algoritmo de Machine Learning, cabe mencionar que también se puede usar para Deep Learning, facilita la adquisición de datos modelos de capacitación, predicciones y refinamiento de resultados, está disponible para el uso en computadores personales, pero es recomendado usarlo en su propio editor en la nube que es Colab.

Keras es un framework de alto nivel para el aprendizaje, escrito en Python y capaz de correr sobre los frameworks TensorFlow, por esta razón se usará para la presente investigación, pues facilita los procesos de experimentación rápida, y ya que al ejecutarse en TensorFlow, por consiguiente se puede ejecutar en Colab.

6. Delimitación

En la presente investigación solamente se utilizará redes neuronales del tipo perceptrón multicapa, donde cabe mencionar que este no es el único tipo de red que existe, i.e., también se tiene Redes Neuronales Convolucionales (RNC), Redes Neuronales Recurrentes (RNR) o bien Redes de Base Radial (RBR) [16]. Así mismo, no se abordará el uso de técnicas d eoptimiación como lo son los algoritmos genéticos, y unicamente se limitará a explorar la mejora en rendimiento al tener más de dos capas duplocadas de pesos en la red con aprendizaje incremental. Así mismo, no se abordarán modelos como las redes profundas u otro conjunto de datos y se limitará el trabajo a lo antes mencionado.

7. Consecuencias

Si el experimento funciona a la perfección ocurrirá lo siguiente:

1. Habrá menos olvido.
2. El aprendizaje tomará menos tiempo.

Como se menciona antes, se van a poder ingresar más datos sin tener que volver a re-entrenar el modelo con todos los datos existentes.

Las tareas de clasificación o predicción podrán ingresar a las RNAs información nueva si tener que entrenar nuevamente el modelo con toda la información historica, reduciendo el cuello de botalle que implica el entrenamiento.

8. Marco Teórico

8.1. Redes Neuronales Artificiales

Las redes neuronales artificiales (RNA) son modelos computacionales de la Inteligencia Artificial los cuales contienen simples unidades de procesamientos llamadas neuronas. Ellas se inspiran en el cerebro humano, tomando como base la conectividad entre neuronas y el aprendizaje que pueden tener. Un perceptron o neurona (artificial) solamente resuelve problemas lineales y tiene la siguiente forma:

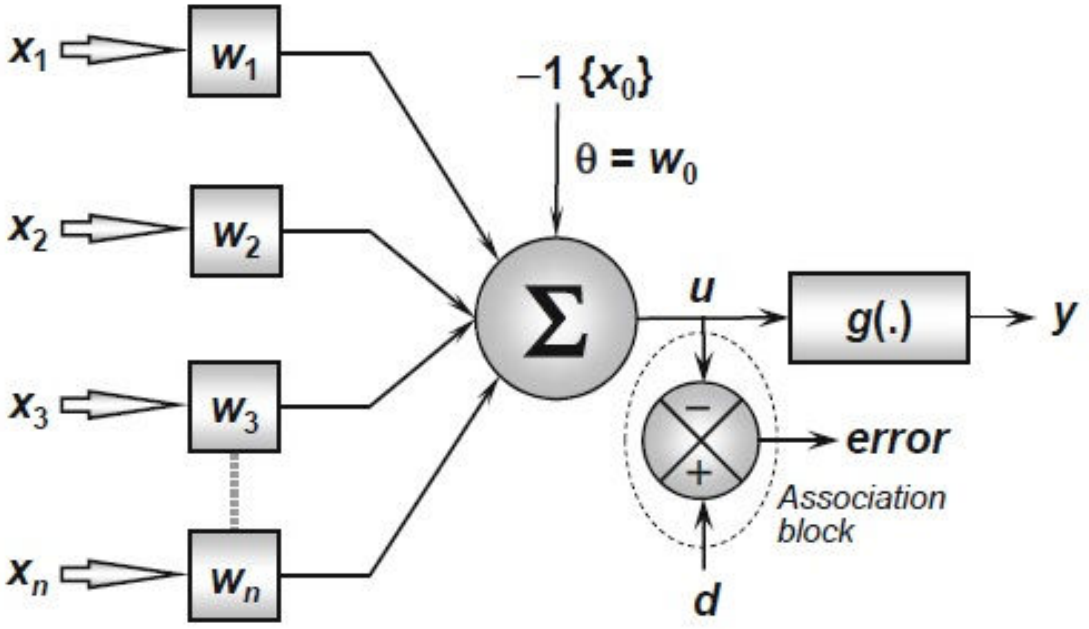


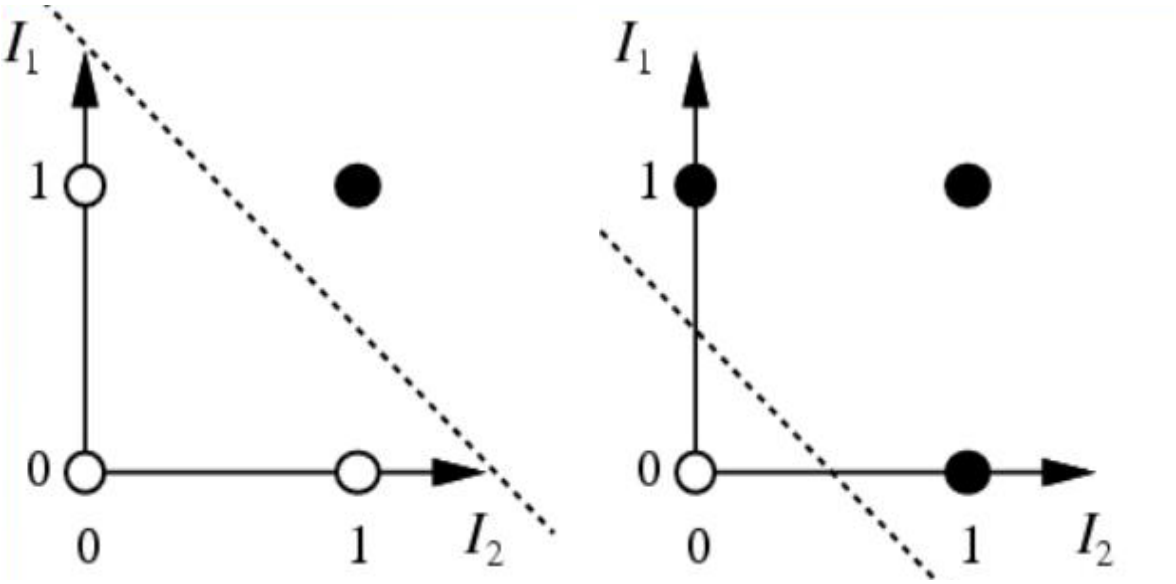
Figura 1: Red Neuronal Artificial Básica

Donde Σ es la representación matemática de la neurona., x_1, x_2, ...,x_n son las variables de entrada a la red. w_1,w_2, ..., w_n son los pesos con los cuales se van a podnerar las entradas, es decir multiplicar cuando la información entra en la neurona. Posterior a multiplicar el peso por la entrada correspondiente, se suman todos esos valores w_1x_1 + w_2x_2 + w_3x_3

Al revisar esta formula, se puede observar que se parece a la operación de una regresión la cual es: y = w_0 + w_ix_i, de esta forma, internamente la neurona realiza una regresión lineal. En su contraparte, el parámetro que permite a la neurona trazar una recta cruzando el eje y en el plano cartesiano (eje de las ordenadas), a ello se conoce como sesgo (del inglés $bias$), este valor se agrega a la conexión, el cual usualmente se le da un valor de 1.

Agregando este nuevo valor a la fórmula, queda de la siguiente manera: $y = \Sigma w_i x_i + w_0 b$, donde *b* es el sesgo.

Un inconveniente del uso de una sola neurona para experimentos es que solo va a resolver ejercicios parecidos a la puerta lógica AND u OR



(a) Puerta Lógica And (b) Puerta Lógica Or

Figura 2: Puertas Lógicas

Pero problemas de tipo XOR no puede, ya que como se nota, una sola neurona sirve para clasificar de un solo lado, así que no puede clasificar ejercicios como los de la Figura (3).

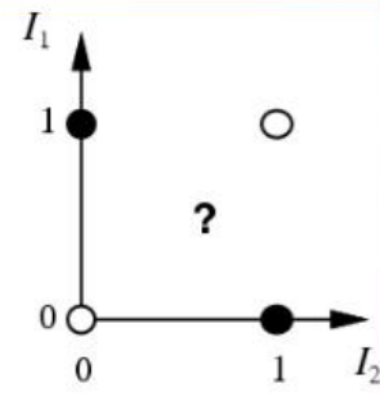


Figura 3: Puerta Lógica Xor

Para solucionarlo se usan dos o más neuronas, además de la función de activación, que es la que permite pasar la informaicón de una neurona a otra, en un rango especificado, y la cual se describirá en la siguiente sección.

8.1.1. Función de Activación

Dicho método se utiliza cuando el modelo de RNA contiene dos o más neuronas. Esta función lo que provoca es dar al modelo una salida no lineal, para eso la segunda fórmula presentada es distorcionada para quedar de la siguiente manera: $f(w_1x_1 + w_2x_2 + w_3x_3 + b_0)$ para el caso de 3 entradas.

Al hablar de funciones de activación se deben de comentar las más comunes, como lo es la función escalonada.

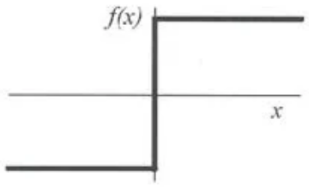


Figura 4: Función Escalonada

Dicha función es representada con:

$$f(x) = \begin{cases} 0 & : x < 0 \\ 1 & : x \geq 0 \end{cases}$$

La función sigmoidal, es una de las más comunes, su forma es:

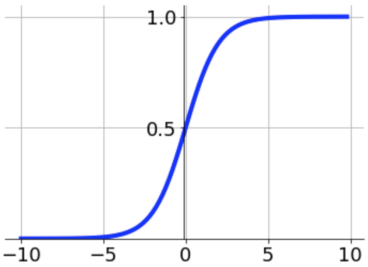


Figura 5: Función Sigmoide

La cual es representada por la siguiente formula:

$$f(x) = \sigma(x) = \left\{ \frac{1}{1 + e^{-x}} \right.$$

Aparte de realizar dicha transformación, las funciones de tranasferencia ayudan en cuestiones probabilísticas ya que se representa del rango de 0 a 1. Aunado a que por ejemplo, la función sigmoidal es diferenciable, lo que permite al algoritmo de backpropagation porde llevar a cabo el entrenamiento.

La función de Unidad Rectificada Lineal o RELU, la cual es una función lineal que cuando es positiva es igual a 1 y cuando es negativa es constante a 0, su forma es:

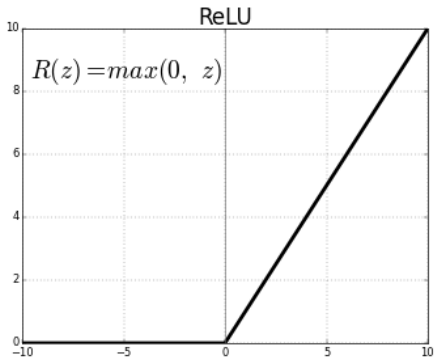


Figura 6: Función ReLU

Se representa con la siguiente formula [7]:

$$f(x) = \begin{cases} 0 & : x < 0 \\ x & : x \geq 0 \end{cases}$$

La función Softmax transforma las salidas a una representación en forma de probabilidades, de tal manera que el sumatorio de todas las probabilidades de las salidas de 1, su gráfica es:

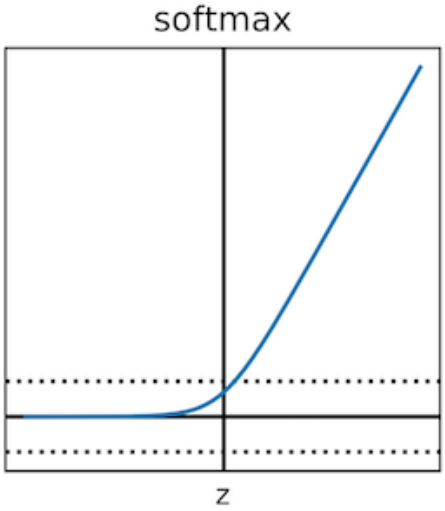


Figura 7: Función Softmax

Su representación matemática es [4]:

$$f(z)_j = \left\{ \frac{e^{z_j}}{\sum_{K=1}^{K} e^{z_k}} \right.$$

Las redes neuronales presentan demasiadas utilidades las cuales ayudan a resolver problemas como los siguientes [12]: no linealidad, mapeo entrada-salida, aprendizaje robusto a errores en los datos de entrenamiento, entre otros. Existen varios tipos de Redes Neuronales tales como: Redes Neuronales de Perceptrón Multicapa, Redes Neuronales Convolucionales, entre otras, las cuales se describirán brevemente más adelante.

8.2. Redes Neuronales de Perceptrón Multicapa

Las Redes Neuronales de Perceptrón Multicapa se pueden dividir en dos capas (las de entrada y salida), pero también en tres o más capas (la de entrada, una o más capas ocultas y la de salida). En las capas ocultas se pueden tener más de una fila de neuronas, las cuales son las encargadas de realizar las operaciones para eliminar la linealidad de los datos. También, como se comentó anteriormente, Sec. (8.1.1) la linealidad de los datos se elimina con las funciones de activación, las cuales modifican los parámetros de la red, permitiendo que se elabore un plano tridimencional, con el cual se puede encontrar la solución al problema planteado.
Además como se explicó anteriormente, no es muy recomendado trabajar con una sola neurona por los problemas presentados al resolver, tareas como el XOR donde se requieren de dos líneas rectas para clasificar el problema correctaemnte.

Como se puede observar en la figura (8), para este caso se cuenta con una MLP que consta de 4 capas, 1 de entrada, 2 ocultas y 1 de salida. En las capas ocultas y de salida se lleva a cabo el procesamiento de las funciones de activación, donde cabe notar que no es así para la primera capa de entrada, donde unicamente sirve para representar las entradas al modelo, i.e. ahí no hay funciones de transferencia.

Así, las neuronas de color azul cuentan con una función (puede ser sigmoidal, escalonada, entre otras), y cuando se llegue a la capa de salida, cada función se va a sumar, de esta manera se obtendrá una función no lineal que de resolución a la tarea a resolver.

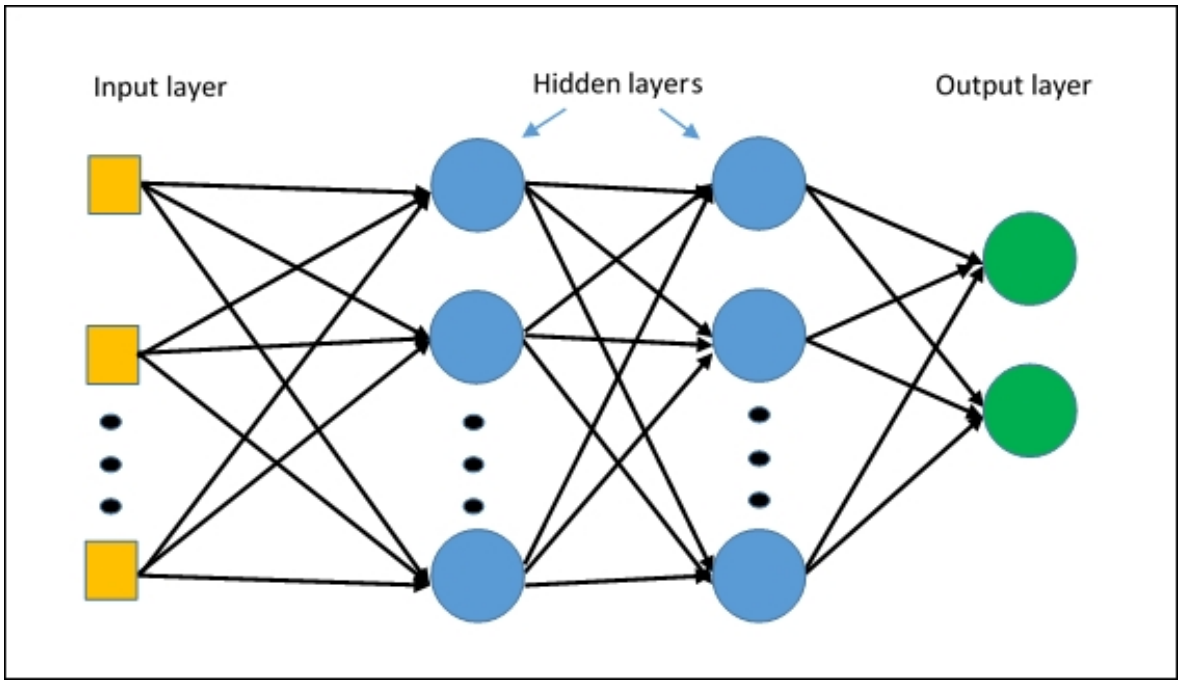


Figura 8: Perceptron Multicapa

8.3. Algoritmo Backpropagation

El bacpkpropagation es un algoritmo de aprendizaje que permite el aprendizaje de la red, permite que una red neuronal pueda auto-ajustar todos sus parámetros para aprender una representación interna de la información que se está procesando. Llegó a dar solución a la limitante del perceptron, este resuelve los problemas lineales, la cual es que no se puede extender a redes más complejas, es decir, a problemas no lineales.

Usando este algoritmo se podrá obtener las derivadas parciales del gradiente y del peso, las cuales sirven para la optimización de la red neuronal, dichas derivadas son: $\frac{\partial C}{\partial W^L}$

Pero no se debe de olvidar que también se debe de calcular las derivadas del sesgo, la derivada anterior queda de la siguiente forma: $\frac{\partial C}{\partial b^L}$ donde *L* pertenece al número de capa donde se encuentra.

Lo que permite a este algoritmo encontrar el error de la derivada es la *chain rule*, la cual en resumen da la siguiente ecuación : $C(a(Z^L))$

Donde Z^L representa el resultado de la suma ponderada, y la forma explicita de Z es: $Z^L = W^L X + b^L$ *a* es la función de activación y *C* es la función de Coste.

Al aplicar la *Chain Rule* se obtiene que las derivadas parciales a conseguir son:

$$\frac{\partial C}{\partial w^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial w^L}$$

$$\frac{\partial C}{\partial b^L} = \frac{\partial C}{\partial a^L} \cdot \frac{\partial a^L}{\partial z^L} \cdot \frac{\partial z^L}{\partial b^L}$$

Como se observa, el uso de todas estas derivadas parciales permiten encontrar el error, en otras palabras, lo que realiza dicho algoritmo es terminar un proceso, si se encuentra un error, este va a regresar hasta la neurona que da error, pero regresa cambiando el valor de *w*, este proceso se va a repetir hasta encontrar el error perfecto, el cual es donde el error disminuye a lo más bajo y el resultado de la red es lo más acertado.

Con lo anterior expuesto, se puede decir que esta metodología es muy útil en el uso de redes neuronales, es por eso que se usa en la investigación [3], para obtener un buen resultado en el aprendizaje incremental, como es explicado ahí es usado para que las redes obtengan una buena topología con buena actualización de pesos.

8.4. Redes Neuronales Convolucionales

Las redes neuronales convolucionales (RNC) son las que facilitan el reconocimiento de imágenes, el trabajo de reconocimiento de imágenes se puede elaborar con RNAs, pero esta tiene demasiadas desventajas, como lo que es pérdida de datos, porque cuando se ingresa una imagen a la capa de entrada, esta se debe de convertir en vectores, si la imagen es de 100 pixeles por 100 pixeles se tendrá un vector de 10000 pixeles, esto solo será en imágenes a blanco y negro, si la imagen es del mismo tamaño pero a color se obtendrá un vector de 30000 pixeles, ya que se usan los filtros RGB (Rojo, Verde, Azul).

Estas redes, presentan una solución para la problemática mencionada anteriormente, se basan en las conexiones que existen en el cerebro junto a la de los ojos.

Sus capas se distribuyen como se muestra Figura (9):

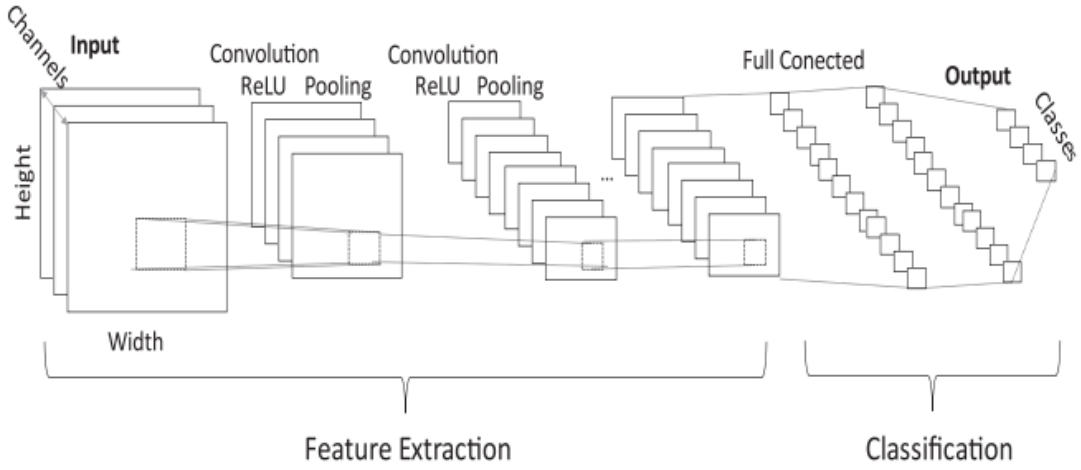


Figura 9: Arquitectura de una RNC

La capa de entrada contiene una matriz o tensor de matriz (más de una matriz), las capas ocultas con dos etapas, y la capa de salida, donde se obtendrá el resultado de la predicción.

La primer etapa es de extracción de características, donde las capas están unidas a capas de *pooling*, estas capas son donde la imagen se divide en distintos kernels (filtros) el cual permitirá hacer submatrices a las imágenes y así poder recrearla en otra matriz, estos filtros deben ser menor de la imagen, se pueden tener más de un kernel para analizar bien la imagen, existen dos métodos de *pooling*: el *Max Pooling* y el *Average Pooling*, el *Max Pooling* extrae el valor máximo de las submatrices para crear otra matriz, como se puede observar en la Figura (10a), mientras que el *Average Pooling* el cual obtiene el promedio de las submatrices y con estos promedios genera otra matriz, como se puede ver en la Figura (10b) para un mejor resultado se tiene una capa de extracción o de convolución con otra capa de pooling, se pueden tener tantas capas de convolución como las de pooling para tener una mejor abstracción de conocimientos, en este tipo de conexiones se usan las funciones de activación conocidas como Relu (8.1.1).

(a) Max Pooling

(b) Average Pooling

Figura 10: Métodos Pooling

La segunda etapa es la de Full Conected, que equivale a tener una red normal o en otras palabras tener una RNA, aquí es donde el modelo va a predecir que es lo que se presenta, puede decir si en la imagen hay un

perro, gato o algún mueble, para eso la función de activación que se usa es softmax (8.1.1), la cual se maneja con probabilidad, esto permitirá que se obtenga una buena probabilidad de lo que se puede obtener por la capa de entrada [5, 14].

8.5. Aprendizaje

8.5.1. Aprendizaje en humanos

El humano tiene una forma de aprendizaje muy particular, la cual se basa del estudio, donde lee, escribe y practica acerca de su tema de interés, pero dicho aprendizaje se puede ir olvidando, esta es una acción muy común que a cualquier persona le sucede. Existen estudios donde se comenta que existen tres motivos del porque se olvidan las cosas, proviene parte de la regularización de las emociones, el como se adquirieron los conocimientos, y porque el olvido es un proceso por el cual el ser humano transita a lo largo de su vida [13]. Pero cabe mencionar que esto no es lo único que causa la perdida de memoria, ya que existe la déficit de memoria.

8.5.2. Aprendizaje Humano

Al momento de hablar del aprendizaje humano, se debe de hablar de la ciencia cognitiva, que es quien se encarga de descubrir esta incógnita, esta ciencia lo estudia de un modo multidisciplinario, el cual abarca las áreas de [2]: antropología, lingüística, filosofía, sicología del desarrollo, ciencia de la computación, neurociencia. Con el método de esta ciencia se pueden descubrir dos tipos de aprendizaje que son: el aprendizaje con compresión y el aprendizaje Activo.

8.5.3. Aprendizaje con Comprensión

La comprensión es una actividad la cual se ha generado al momento de realizar cualquier tipo de lectura.

Teniendo un enfocamiento en el ámbito estudiantil, ya que es donde más se maneja esta táctica, esto es una practica algo compleja, sistemática y organizada, pues da el significado de la literatura y durabilidad del aprendizaje. Al conocer esto se puede decir con seguridad que para cualquier tipo de aprendizaje la comprensión es una parte primordial [15].

8.5.4. Aprendizaje Activo

El aprendizaje de la forma en la que se conoce no es del todo efectiva, ya que el sistema educativo no se basa en el principio de *belongingness*, el cual esta asociado al estimulo con su respuesta, y esto es lo más importante para que el ser humano pueda aprender cualquier cosa.

Este tipo de aprendizaje se basa en la recepción de conocimientos y la práctica donde se ponen en marcha los conocimientos adquiridos.
Otro concepto importante aquí es la tautología doble (*selbsttätiges Lernen*), que en palabras informales es convertirse en autodidacta, se puede observar que esto pertenece a dicho aprendizaje, porque usa el principio mencionado anteriormente [10].

8.5.5. Aprendizaje Incremental

Con el pasar de los años la tecnología a evolucionado, eso quiere decir que el Aprendizaje Automático se ha actualizado y que la cantidad de datos va aumentado con más frecuencia.

Se puede verificar como *Üna tarea de aprendizaje es incremental si los ejemplos de entrenamiento usados para resolverla están disponibles en horas extras, generalmente uno a la vez"* [9], si los resultados no se necesitan de manera urgente, este tipo de trabajos serán resueltos por algoritmos de aprendizaje no incremental.

Una área donde esto es de mucha utilidad es la *Robótica* porque este necesita estar en constante entrenamiento [9].

Dicha forma de aprender fue inspirada en la forma en que el humano aprende y esta más rápida, fue por esto que fue adoptada por el aprendizaje máquina.

Con el paso del tiempo se ha convertido en un paradigma del aprendizaje automático, aquí el aprendizaje toma el lugar de nuevos ejemplos para juntarlos y conforme van aprendiendo estos toman el lugar de los ejemplos ya aprendidos [12].

8.5.6. Algoritmos de Aprendizaje Incremental

El algoritmo de aprendizaje incremental puede definirse como aquel que cumple los siguientes criterios: 1) Ser capaz de aprender y actualizarse con cada nuevo dato etiquetado o no etiquetado. 2) Conservar los conocimientos adquiridos previamente. 3) No debe requerir el acceso a los datos originales. 4) Generar una nueva clase o cluster cuando sea necesario. Dividir o fusionar los clusters cuando sea necesario. 5). Ser de naturaleza dinámica con el entorno cambiante [1].

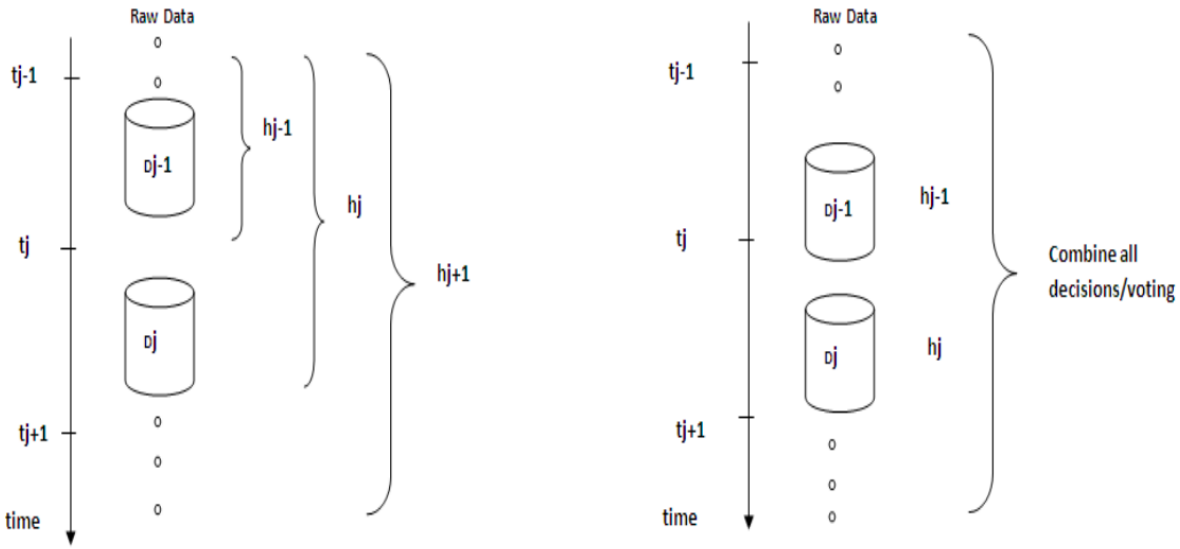


Figura 11: Dos enfoques tradicionales del aprendizaje incremental.

1 Metodología de acumulación de datos. 2 Metodología de aprendizaje por conjuntos.

Como se observa en la Figura 11, en el primer método, cuando se recibe una nueva porción de datos Dj, se descarta hj-1 y se desarrolla una nueva hipótesis hj, basada en todos los datos disponibles acumulados hasta el momento. Y en el segundo método, cuando se recibe una nueva porción de datos Dj, se desarrolla una única hipótesis nueva o un conjunto de hipótesis nuevas basadas en los nuevos datos. Por último, se puede utilizar un mecanismo de votación para combinar todas las decisiones de las diferentes hipótesis y obtener la predicción final.

Por ejemplo, si se deja que Dj-1 represente la porción de datos recibida entre el tiempo tj-1 y tj, y que la hipótesis hj-1 se desarrolle sobre Dj-1.
El sistema aprenderá información de forma adaptativa cuando se reciba una nueva porción de datos Dj. En el método de aprendizaje por conjuntos, se desarrolla una nueva hipótesis hj o un conjunto de hipótesis H:h1, i1,2,...,M, basadas en los nuevos datos. A continuación, se utiliza el mecanismo de votación para combinar todas las decisiones de las diferentes hipótesis y llegar a la predicción final. La mayor ventaja de este enfoque es que no se requiere almacenar los datos vistos anteriormente, el conocimiento se ha almacenado en la serie de hipótesis desarrolladas a lo largo de la vida de aprendizaje.

Conocimiento en el momento t:
Dt es un trozo de datos con n instancias (i=1,...,n)
(xi,yi) es una instancia en el espacio de características m-dimensional X
Yi ∈ Y =1,...,K clases
Función de distribución Df
Una hipótesis ht, desarrollada por los datos basados en Dt con Pt
La nueva entrada estará disponible en el momento (t+1)

Algoritmo de aprendizaje:

1. Encontrar la relación entre Dt y Dt+1

2. Actualizar la función de distribución inicial Dt+1
3. Aplicar la hipótesis ht a Dt+1 y calcular el pseudoerror de ht
4. Refinar la función de distribución para Dt+1
5. Se desarrolla una hipótesis por los datos basados en Dt+1 con Pt+1
6. Repetir el procedimiento cuando se reciba la siguiente porción del nuevo conjunto de datos.

Resultado: La hipótesis final.

Ün algoritmo de aprendizaje es incremental si, para cualquier muestra de entrenamiento dada:

$$e_1, ..., e_s$$

produce un secuencia de hipótesis

$$h_0, h_1, ..., h_n$$

tal que

$$h_{i+1}$$

depende solo de

$$h_i$$

y del ejemplo actual e" [9], como se observa, estos son algoritmos que permiten a la inteligencia artificial poder realizar actividades de predicción de una manera más eficaz.

Un ejemplo del uso de esta rama es el proyecto *COBWEB*, donde se trata de categorizar el número de Clúster y la pertenencia de dichas categorías por medio de una métrica probabilística global, esto lo realiza por medio de que se agrega una nueva categoría, este proceso lo que realizará es actualizar todas las probabilísticas con los nuevos datos recabados [8].

9. Metodología

El primer paso a realizar, es recrear el código mostrado en [3], el describe la implementación de una red neuronal multicapa usando el algoritmo de entrenamiento backpropagation en el lenguaje de programación Python. Así mismo, se utilizará el conjunto de datos de Optical Digits, en donde se tendrá que preprocesar los datos, para eliminar registros inválidos.

Posteriormente, se implementará una extensión del código, donde se experimentará con mas de dos capas de pesos duplicados para mejorar la tasa de olvido de información al momento de usar el aprendizaje incremental. Para ello se explorará incrementando gradualmente el número de capas de pesos duplicados.

Finalmente, cuando los resultados se obtengan se realizará una comparación, de los resultado del algoritmo base con los resultados del algoritmo extendido.

10. Cronograma de Actividades

	PRIMER SEMESTRE						SEGUNDO SEMESTRE					
	MESES											
FASES	1	2	3	4	5	6	1	2	3	4	5	6
Delimitación del tema												
Preprocesamiento de datos												
Implementación del algoritmo												
Extensión del algoritmo												
Comparación de ambas implementaciones												
Preparación del trabajo para congresos												
Escritura de tesis												
Pre-examen												
Atender observaciones												
Examen profesional												

11. Organización del Capitulado

En el capitulo 2 se explicará lo que son y como funcionan las redes neuronales artificiales, asıcomo la función de activación que es un método que utilizan estas. Se describirán los tipos de redes neuronales, tanto de perceptron multicapa como convolucionales, y el algoritmo backpropagation. Se mencionará como es el aprendizaje en humanos, que este se divide en el aprendizaje activo y el aprendizaje con comprensión Y para terminar se describirá el aprendizaje incremental y su algoritmo.

En el capitulo 3 se implementará el algoritmo de John A. Bullinaria, se verificará su funcionamiento y los resultados que da al pasar los datos que dice para comprobar que es como menciona en su artículo.
En el capitulo 4 se explicará como se se extendió el algoritmo base permitiendo el uso de mas de dos pesos duplicados y se aplicaran los mismos datos de entrenamiento y de prueba que al algoritmo base.

Posteriormente, en el capitulo 5 con los resultados obtenidos, se mostrará una comparación de los resultados de ambos trabajos para notar si hubo una reducción significativa en las tasas de aprendizaje. En el capitulo 6 se verán las conclusiones y trabajo futuro.

Referencias

[1] RR Ade and PR Deshmukh. Methods for incremental learning: a survey. *International Journal of Data Mining & Knowledge Management Process*, 3(4):119, 2013.

[2] J Bransford, A Brown, and R Cocking. Cómo aprende la gente: cerebro, mente, experiencia, y escuela. *Revista del Instituto de Matemática y Física*, pages 44–64, 2000.

[3] John A Bullinaria. Evolved dual weight neural architectures to facilitate incremental learning. In *IJCCI*, pages 427–434, 2009.

[4] Calvo Diego. Función de activación – Redes neuronales, 12 2018.

[5] Jaime Durán Suárez. Redes neuronales convolucionales en r: Reconocimiento de caracteres escritos a mano. 2017.

[6] Ryan Elwell and Robi Polikar. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531, 2011.

[7] Sarahí Silva Estefanía Freire. Redes neuronales - Bootcamp AI, 12 2021.

[8] Douglas H Fisher. Knowledge acquisition via incremental conceptual clustering. *Machine learning*, 2(2):139–172, 1987.

[9] Christophe G. Giraud-Carrier. A note on the utility of incremental learning. *AI Commun.*, 13:215–224, 2000.

[10] Günter L. Huber. Aprendizaje activo y metodologías educativas. 2008.

[11] Ai-Jun Li. An improved algorithm for incremental learning learn++. In *2008 International Conference on Wavelet Analysis and Pattern Recognition*, volume 1, pages 310–315. IEEE, 2008.

[12] Yuan Liu. Yuan liu incremental learning in deep neural networks. 2015.

[13] Simon Nørby. Why forget? on the adaptive value of memory loss. *Perspectives on Psychological Science*, 10:551 – 578, 2015.

[14] Flor G. Ortiz-Gomez, Daniele Tarchi, Ramón Martínez, Alessandro Vanelli-Coralli, Miguel A. Salas-Natera, and Salvador Landeros-Ayala. Convolutional neural networks for flexible payload management in vhts systems. *IEEE Systems Journal*, 15(3):4675–4686, 2021.

[15] Karel Pérez Ariza and José Emilio Hernández Sánchez. Aprendizaje y comprensión. una mirada desde las humanidades. *Humanidades Médicas*, 14(3):699–709, 2014.

[16] Paloma Royo. Qué son las redes neuronales y cuál es su aplicación en el marketing, 2021.

[17] Luis A Cruz Salazar, David J Muñoz Aldana, and Juan A Contreras Montes. Implementación de redes neuronales y lógica difusa para la clasificación de patrones obtenidos por un sónar. In *2013 II International Congress of Engineering Mechatronics and Automation (CIIMA)*, pages 1–6. IEEE, 2013.

[18] Beibei Zhao, Tairan Wu, Fang Fang, Lin Wang, Wenzhang Ren, Xu Yang, Zhangjing Ruan, and Xuejin Kou. Prediction method of 5g high-load cellular based on bp neural network. In *2022 8th International Conference on Mechatronics and Robotics Engineering (ICMRE)*, pages 148–151. IEEE, 2022.