UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MÉXICO

CENTRO UNIVERSITARIO UAEM VALLE DE MÉXICO

Desarrollo de un Modelo de Aprendizaje Incremental de Redes Nueronales Artificiales Aplicado al Conjunto de Datos Optdigit

PROTOCOLO DE INVESTIGACIÓN

P r e s e n t a

C. González Hernández Luis Ángel

Asesor: Dr. Víctor Manuel Landassuri Moreno
Co-Asesora: Ing. Carmen Lucía Bustillo Hernández

Atizapán de Zaragoza, Edo. de Méx. Enero 2025

Centro Universitario
UAEM Valle de México

Índice

Índice de figuras

Desarrollo de un Modelo de Aprendizaje Incremental de Redes Neuronales Artificiales Aplicado al Conjunto de Datos Optdigit

24 de febrero de 2025

Resumen

En un enfoque normal, primero se entrena el modelo con un conjunto de datos inicial, y una vez que el modelo está listo, se utiliza para realizar predicciones. Sin embargo, si llega nueva información, es necesario reentrenar el modelo desde cero, utilizando tanto la información histórica como la nueva. Este proceso puede ser computacionalmente costoso y poco eficiente, especialmente cuando se manejan grandes volúmenes de datos.

El aprendizaje incremental es un área de la Inteligencia Artificial que 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 [2]. En el presente trabajo de investigación se utilizará el modelo de Redes Neuronales Artificiales enfocado en la clasificación de dígitos escritos a mano, empleando el algoritmo de entrenamiento de backpropagation, con redes Multicapa Perceptron y duplicación de pesos múltiples, simulando memoria a corto y largo plazo para mejorar los resultados presentados en [2].

Palabras clave: Aprendizaje incremental, Redes Neuronales Artificiales, Clasificación de dígitos.

1. Introducción

La inteligencia artificial (IA) es un campo del conocimiento que busca desarrollar máquinas capaces de emular el comportamiento y razonamiento humano, permitiendo una interacción casi indistinguible de la que se tendría con otro ser humano. Esta área no solo se enfoca en la creación de sistemas inteligentes, sino también en desarrollar herramientas que faciliten y optimicen las actividades cotidianas.

Un subcampo de la IA es el Aprendizaje Automático (Machine Learning), que estudia algoritmos capaces de aprender tareas de manera autónoma. Dentro de este campo, las Redes Neuronales Artificiales (RNAs) son una de las técnicas más reconocidas, pues utilizan procesos matemáticos para resolver tareas complejas. Las RNAs han demostrado ser útiles en áreas como la predicción de capacidades de redes 5G basadas en el tráfico diario [11], así como en la clasificación de materiales como metales y rocas mediante lógica difusa [10].

Las RNAs están conformadas por neuronas artificiales que simulan las biológicas. En este contexto, los procesos químicos que ocurren en el cerebro se imitan computacionalmente a través de señales que viajan entre las neuronas artificiales, que en adelante llamaremos ”neuronas”. Esta estructura distribuida y paralela otorga a las RNAs una notable capacidad de aprendizaje [7].

En el contexto del aprendizaje automático, cuando una técnica como las RNAs se enfoca en aprender una tarea específica, se considera un *algoritmo lineal sin memoria*. Desde sus inicios, este ha sido uno de los enfoques más utilizados en las RNAs [5]. Sin embargo, cuando se necesita incorporar nueva información sin reentrenar todo el modelo, surge el concepto de Aprendizaje Incremental, que busca incluir nuevos datos sin tener que entrenar el modelo desde cero.

Un aspecto derivado del aprendizaje incremental es el concepto de memoria en la IA, que explora cómo un algoritmo de aprendizaje automático puede olvidar información previamente adquirida al incorporar nuevos datos. Esta problemática, análoga a la pérdida de memoria humana, plantea desafíos tanto para máquinas como para seres humanos. En respuesta, se han diseñado diversos enfoques para mitigar este fenómeno. Un ejemplo es el trabajo de [2], que propone el uso de RNAs con pesos dobles, donde la primera capa de pesos se comporta como memoria a corto plazo y la segunda como memoria a largo plazo. Los experimentos realizados en [2] muestran una mejora en tareas de aprendizaje incremental, reduciendo la pérdida de información en comparación con implementaciones previas como el algoritmo Learn++ [6, 3].

Este trabajo de investigación tiene como objetivo explorar nuevas configuraciones de pesos duplicados, ampliando y extendiendo los resultados obtenidos en [2].

2. Planeamiento del Problema

Las Redes Neuronales Artificiales (RNAs), en especial las *Multilayer Perceptrons* (MLP), tienen la capacidad de aprender tareas y resolver problemas de predicción y clasificación. Utilizando algoritmos de aprendizaje como el *Backpropagation*, es posible ajustar los pesos de una RNA para que pueda abordar una tarea específica. Las MLP son una forma común de RNA, compuestas por múltiples capas de neuronas, donde la capa de entrada recibe los datos y la capa de salida proporciona las predicciones. Las capas intermedias, también conocidas como capas ocultas, realizan transformaciones no lineales que permiten a la red aprender representaciones complejas de los datos. Sin embargo, muchas de las tareas que se resuelven en la vida cotidiana generan información adicional con el tiempo. Un ejemplo de esto es el comportamiento de una serie financiera o la predicción del clima en una determinada región. En este contexto, surge el aprendizaje incremental, un enfoque poco explorado que permite que el modelo aprenda nueva información sin la necesidad de reentrenar todo el modelo con los datos antiguos y nuevos.

En los modelos actuales de aprendizaje automático, cuando se utiliza un conjunto de datos para entrenar un modelo específico, dicho modelo es funcional solo para ese conjunto y la información que representa. Sin embargo, al ser necesario incorporar nueva información al modelo, se debe recolectar esa nueva información, añadirla a la ya existente y luego reentrenar todo el modelo para integrar los datos nuevos.

En este contexto, si una red entrenada con un primer conjunto de datos d_1 debe entrenarse con un segundo conjunto de datos d_2, al hacerlo, se perderá el conocimiento adquirido por d_1. Si no se utiliza un modelo de aprendizaje incremental, se debe combinar d_1 y d_2 en un solo conjunto y reentrenar la RNA para incorporar el nuevo conocimiento (d_2). En cambio, si se emplea el aprendizaje incremental, la RNA se entrena inicialmente con d_1 y luego con d_2, manteniendo una mínima pérdida de información de d_1. Si en el futuro se obtiene más información del problema (d_3), se entrenará la RNA con d_3 usando el modelo incremental, minimizando la pérdida de datos de d_1 y d_2.

Este trabajo se basa en la investigación de [2], que utiliza una configuración de pesos duplicados en la RNA. En esta configuración, los pesos de la RNA se duplican y se asignan diferentes tasas de aprendizaje a las capas. La primera capa, asociada a una tasa de aprendizaje alta, simula la memoria a corto plazo al aprender rápidamente una nueva tarea y olvidar más rápido la información anterior. La segunda capa, con una tasa de aprendizaje baja, simula la memoria a largo plazo, aprendiendo más lentamente y olvidando menos la información previa. Al integrar ambas capas, la RNA pondera la salida para considerar tanto la nueva información adquirida como la olvidada en ambas capas de pesos.

El problema principal del aprendizaje incremental en [2] es que, a medida que se incorporan nuevos conjuntos de datos, se pierde progresivamente el conocimiento adquirido de los primeros conjuntos, lo cual se vuelve menos útil si en el futuro se tienen 10 o 20 etapas de entrenamiento incremental con nuevos datos.

Por lo tanto, es crucial explorar nuevas configuraciones de RNAs que mejoren los métodos actuales para reducir la cantidad de información olvidada a medida que llega nueva información. Al igual que en investigaciones anteriores, este trabajo se basará en los conceptos de memoria a corto y largo plazo. Sin embargo, en lugar de duplicar los pesos y tener dos tasas de aprendizaje, se propondrá la idea de crear más copias de los pesos, cada una con una tasa de aprendizaje distinta.

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 [2] 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 [2].

3. Separar el conjunto de entrenamiento y de prueba de acuerdo a lo que se explica en el artículo [2] y probar el primer código implementado en miras de comprobar los resultados previamente mostrados en [2].

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

La inclusión de más de dos capas de pesos duplicados, cada una con sus respectivas tasas de aprendizaje, puede reducir el olvido de la información previamente aprendida en un modelo de aprendizaje incremental utilizando redes neuronales artificiales del tipo MLP, cuando se aplica el algoritmo de Backpropagation.

5. Justificación

Las redes neuronales artificiales permiten el aprendizaje automático y la resolución de distintos problemas. Sin embargo, como se mencionó anteriormente, las técnicas de aprendizaje automático presentan una deficiencia importante: cuando se incorporan nuevos bloques de datos, se observa un deterioro en el rendimiento del aprendizaje y un olvido de la información previamente aprendida [2].

Aunque los resultados obtenidos hasta ahora no han sido perfectos, la memoria a corto plazo de las redes neuronales tiende a olvidar con el tiempo, lo cual es una limitación. En contraste, los seres humanos pueden aprender nuevas tareas o información de un problema sin olvidar de manera significativa lo que previamente aprendieron. Aunque biológicamente los humanos tienen una estructura cerebral que les permite aprender y retener mejor la información, actualmente no existe ningún procedimiento que permita modificar la estructura del cerebro para mejorar la retención y reducir el olvido.

Desde un punto de vista computacional, no hay limitaciones inherentes que impidan experimentar con nuevas configuraciones de redes neuronales artificiales (RNAs) para lograr que toda la información ingresada en el modelo se acumule, sin problemas de almacenamiento, y sin olvidar la información previa. Esto podría ser beneficioso en diversas aplicaciones.

En un escenario en el que no se utilice aprendizaje incremental, la llegada de nueva información implicaría la necesidad de volver a entrenar todo el modelo con la información anterior y la nueva. Esto sería un proceso costoso en términos de cómputo y energía, ya que el entrenamiento de una RNA es uno de los cuellos de botella principales, lo que significa un alto consumo de recursos. En cambio, si solo se entrena con la nueva información, se podrían reducir tanto el tiempo como el consumo energético.

Existen varias herramientas que permiten codificar redes neuronales artificiales utilizando librerías preexistentes. En esta investigación, se destacan dos plataformas principales: Microsoft Azure y Google Colab. Azure ofrece una máquina virtual para programar en Python, mientras que Google Colab proporciona un entorno similar, pero de forma gratuita. Ambas herramientas permiten trabajar con Python, un lenguaje de programación libre y fácil de depurar, ideal para el desarrollo de aplicaciones de Machine Learning.

Una de las librerías más populares para Machine Learning es TensorFlow, que facilita la creación y entrenamiento de redes neuronales, permitiendo detectar patrones y razonamientos. Debido a sus capacidades y a su compatibilidad con Deep Learning, TensorFlow será la librería utilizada en esta investigación, particularmente porque es compatible con Keras, un framework de alto nivel que se ejecuta sobre TensorFlow. Keras simplifica los procesos de experimentación rápida y es ideal para su ejecución en plataformas como Google Colab.

6. Delimitación

En la presente investigación, se utilizarán exclusivamente redes neuronales del tipo perceptrón multicapa (MLP). Cabe señalar que este no es el único tipo de red neuronal existente, ya que también existen Redes Neuronales Convolucionales (RNC), Redes Neuronales Recurrentes (RNR) y Redes de Base Radial (RBR) [9]. Además, no se abordarán técnicas de optimización como los algoritmos genéticos, limitándose únicamente a explorar la mejora en el rendimiento de redes neuronales con más de dos capas duplicadas de pesos en un contexto de aprendizaje incremental. Asimismo, el estudio se restringe a las redes mencionadas, sin incluir modelos de redes neuronales profundas ni el uso de otros conjuntos de datos.

7. Consecuencias

Si el experimento tiene éxito, se espera que ocurran las siguientes consecuencias:

1. Se reducirá el olvido de la información previamente aprendida.

2. El tiempo de aprendizaje será menor.

Como se mencionó previamente, será posible incorporar nuevos datos sin necesidad de reentrenar el modelo con toda la información existente.

Las tareas de clasificación o predicción podrán integrar información nueva en las redes neuronales artificiales sin la necesidad de volver a entrenar el modelo con todos los datos históricos, lo que permitirá reducir el cuello de botella asociado al proceso de entrenamiento.

8. Marco Teórico

8.1. Optical Digit

Es un conjunto de datos utilizado en el área de reconocimiento de patrones, el motivo de su creación fue para la investigación de reconocimiento de caracteres ópticos de caracteres (OCR). Su representación es de imagenes en la escala de grises con un tamaño de 8x8 pixeles (Como se muestra en la Figura 1), cada imagen representa un número en el rango de 0 a 9,Su representación es de imágenes en la escala de grises con un tamaño de 8x8 píxeles (como se muestra en la Figura 1), cada imagen representa un número en el rango de 0 a 9, dicho dataset contiene un total de 5620 imágenes, apesar de esto el dataset tiene un balance de clases, lo que quiere decir que cada clase (en este caso cada dígito) tiene la misma cantidad de muestras.

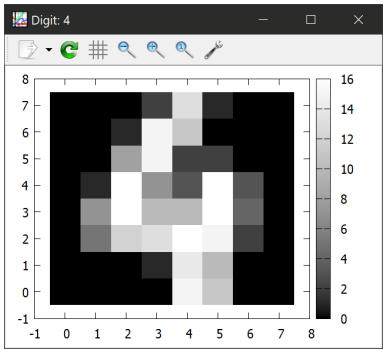


Figura 1: Conjunto de Datos Optical Digit

Este conjunto de datos se utiliza en diferentes áreas de la inteligencia artificial, tales como:

1. Clasificación de digitos.
2. Reconocimiento de patrones.
3. Comparación de modelos.
4. Aprendizaje supervisado.

Al usar este tipo de conjunto de datos se obtienem algunas ventajas que por su simplicidad y tamaño permite que los test sean más rápidas que con otros.

8.2. Redes Neuronales Artificiales

Las redes neuronales artificiales (RNA) son modelos computacionales dentro de la Inteligencia Artificial que contienen unidades de procesamiento simples llamadas neuronas. Estas se inspiran en el cerebro humano, basándose en la conectividad entre neuronas y el aprendizaje que pueden tener. Un perceptrón o neurona (artificial) solo resuelve problemas lineales y tiene la siguiente forma:

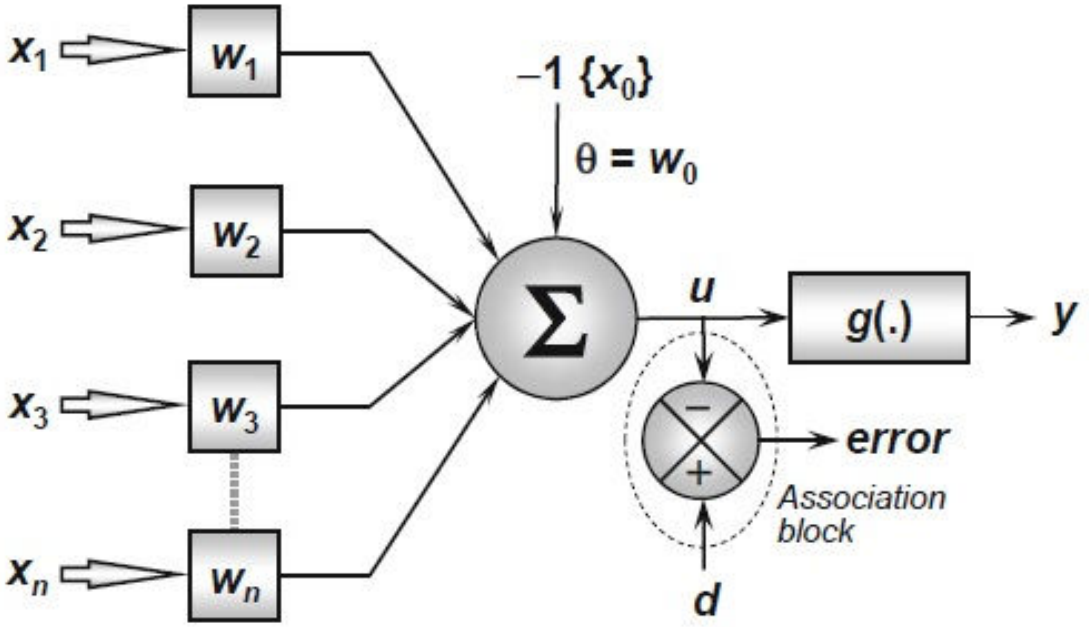


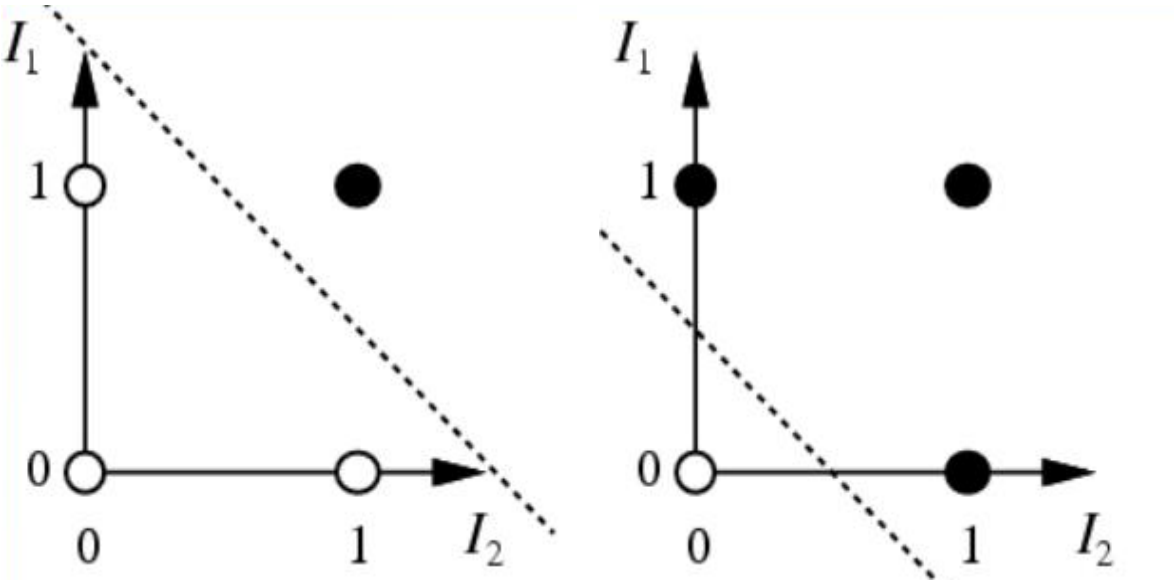
Figura 2: Red Neuronal Artificial Básica

Donde Σ es la representación matemática de la neurona. x_1, x_2, ...,x_n son las variables de entrada a la red, y w_1, w_2, ...,w_n son los pesos con los cuales se ponderan las entradas, es decir, se multiplican cuando la información entra en la neurona. Posteriormente, se suman todos esos valores: $w_1x_1 + w_2x_2 + w_3x_3$.

Al revisar la fórmula anterioir, se puede observar que se parece a la operación de una regresión, la cual es: $y = w_0 + w_ix_i$. De esta manera, internamente la neurona realiza una regresión lineal. El parámetro que permite a la neurona trazar una recta cruzando el eje y en el plano cartesiano (eje de las ordenadas) es conocido como sesgo (del inglés $bias$). Este valor se agrega a la conexión y usualmente se le asigna 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.

Sin embargo, se debe de tener presente que al usar una sola neurona, la única problemática que se puede resolver, son los problemas lineales, un ejemplo de esto es la resolución de problemas de puertas lógicas de tipo AND u OR, las cuales se presentan en la Figura 3, si se necesita la resolución de problemas no lineales o también conocidos como puerta lógica de tipo XOR, Figura 4 no podrá, esto sucede ya que al ser un problema de tipo lienal no puede separar de manera correcta los datos que se le asigna, en cambio una resolución de problemas no lineales como lo es la puerta XOR permite realizar una correcta agrupación de clases (0 y 1), para poder realizarlo se debe de implementar las redes neuronales de multicapa, mejor conocidas como *Perceptron Multicapa* las cuales permiten combinar n numero de capas neuronales.



(a) Puerta Lógica AND (b) Puerta Lógica OR

Figura 3: Puertas Lógicas

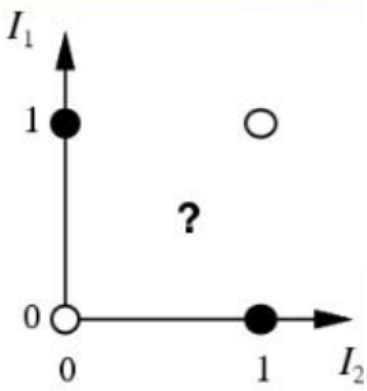


Figura 4: Puerta Lógica XOR

Además del uso de 2 o más capas dentro de la neurona, es indispensable el uso de una función de activación (Sección 8.2.1) que permite pasar la información de una neurona a otra dentro de un rango específico, lo cual se describirá en la siguiente sección.

8.2.1. Función de Activación

Este método se utiliza cuando el modelo de RNA contiene dos o más neuronas, además proporciona al modelo una salida no lineal. Para este tipo de problemáticas donde tienen 3 entradas, se utiliza la siguiente fórmula: $f(w_1x_1 + w_2x_2 + w_3x_3 + b_0)$. Sin mencionar que las funciones de transferencia ayudan en cuestiones probabilísticas, ya que se representan en un rango de 0 a 1 [8].

Al hablar de funciones de activación, se deben comentar las más comunes:

- **Función Escalonada:**
  Esta función se utiliza para problemas de clasificación, ya que su salida es binaria, es decir, 0 o 1.

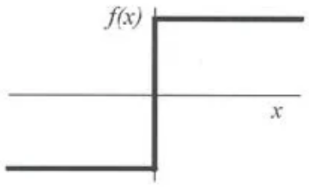


Figura 5: Función Escalonada

Dicha función está representada por:

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

- **Función Sigmoide:**
  Esta función es una de las más comunes, ya que su salida es un rango de 0 a 1, lo que permite interpretarla como una probabilidad.

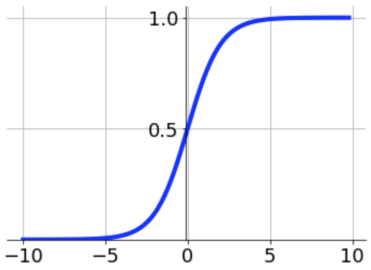


Figura 6: Función Sigmoide

Está representada por la siguiente fórmula:

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

Sin mencionar que este tipo de funciones es ajustable, lo cual es una característica importante del algoritmo de retropropagación (Sección: 8.4).

- **Función de Unidad Rectificada Lineal (ReLU):**
  Esta función es la más utilizadas en RNAs ya que supera los problemas de desvanecimeinto del gradiente, además de ser más rápida en el entrenamiento.
  Es una función lineal que, cuando es positiva, toma el valor de la entrada, y cuando es negativa, toma el valor de 0.

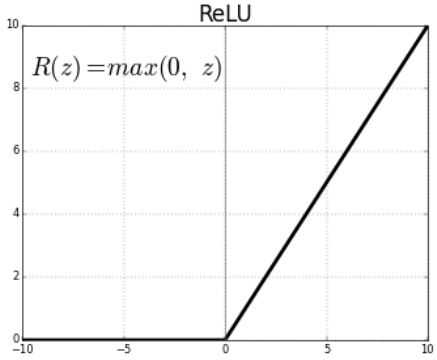


Figura 7: Función ReLU

Está representada por la siguiente fórmula:

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

- **Función Softmax:**
  Esta función se utiliza en la capa de salida de la red neuronal, ya que transforma las salidas en una representación probabilística, de tal manera que la suma de todas las probabilidades sea 1.

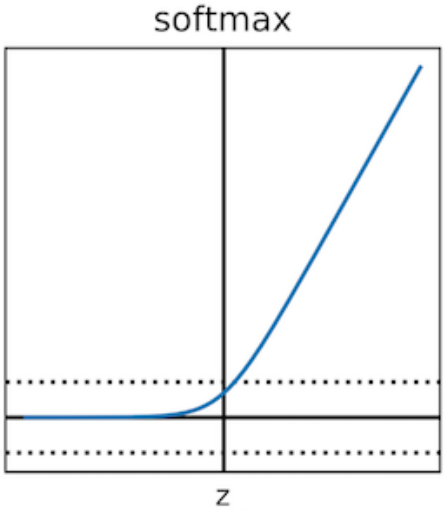


Figura 8: Función Softmax

Su representación matemática es:

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

- **Función Tangente Hiperbólica:**
  Esta función es similar a la función sigmoide, pero su rango es de -1 a 1. También sufre de problemas de desvanecimiento del gradiente, lo que puede dificultar el entrenamiento de redes neuronales profundas.

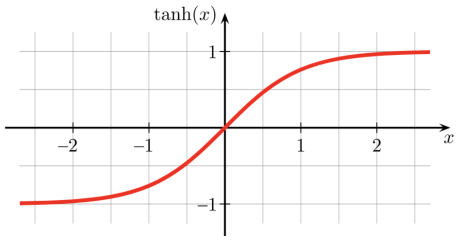


Figura 9: Función Tangente Hiperbólica

Su representación matemática es:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

Las redes neuronales presentan diversas utilidades que ayudan a resolver problemas como no linealidad, mapeo entrada-salida, aprendizaje robusto a errores en los datos de entrenamiento, entre otros. Existen varios tipos de redes neuronales, como las redes neuronales de perceptrón multicapa y redes neuronales convolucionales, que se describirán brevemente más adelante.

8.3. Redes Neuronales de Perceptrón Multicapa

Las Redes Neuronales de Perceptrón Multicapa (MLP) pueden dividirse en dos capas (entrada y salida), pero también pueden tener tres o más capas (entrada, una o más capas ocultas y salida). En las capas ocultas, se pueden tener más de una fila de neuronas, que son las encargadas de realizar las operaciones para eliminar la linealidad de los datos. Como se comentó anteriormente en la sección 8.2.1, la linealidad de los datos se elimina con las funciones de activación, que modifican los parámetros de la red, permitiendo la elaboración de un plano tridimensional con el cual se puede encontrar la solución al problema planteado.

Además, como se explicó anteriormente, no es recomendable trabajar con una sola neurona debido a los problemas al resolver tareas como XOR, donde se requieren dos líneas rectas para clasificar el problema correctamente.

Como se puede observar en la Figura 10, para este caso se cuenta con una MLP que consta de 4 capas: una de entrada, dos ocultas y una de salida. En las capas ocultas y de salida se lleva a cabo el procesamiento de las funciones de activación, mientras que en la capa de entrada no se aplica ninguna función de transferencia, ya que simplemente representa las entradas al modelo.

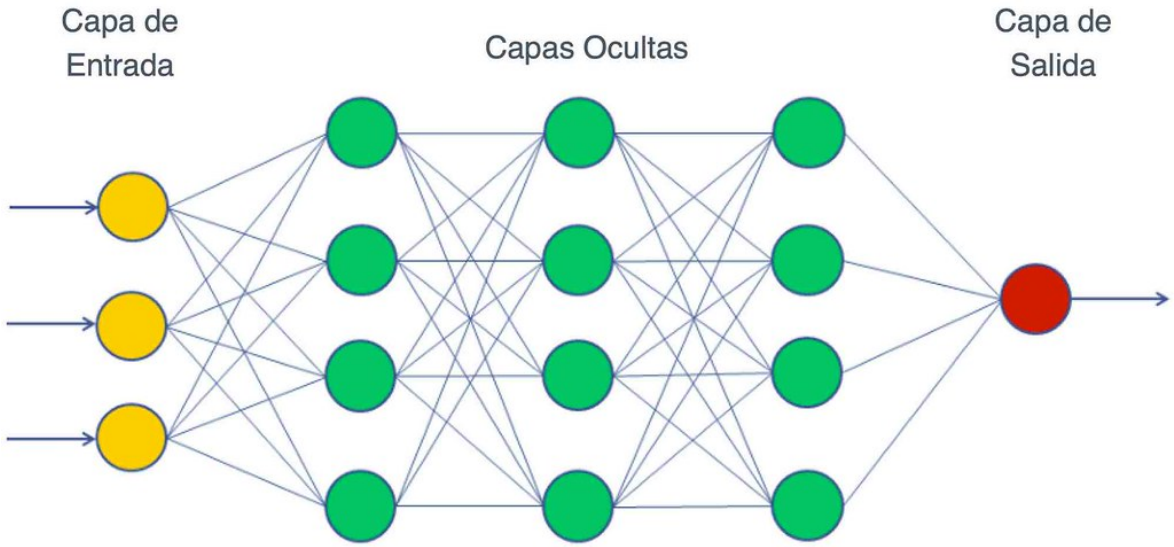


Figura 10: Red Neuronal Multicapa

8.4. Algoritmo de Retropropagación

El algoritmo de retropropagación es un algoritmo de aprendizaje que permite que una red neuronal autoajuste todos sus parámetros para aprender una representación interna de la información que está procesando. Llegó para resolver la limitante del perceptrón, que solo resuelve problemas lineales, y se extiende a redes más complejas, es decir, a problemas no lineales.

Mediante este algoritmo se obtienen las derivadas parciales del gradiente y los pesos, los cuales se utilizan para optimizar la red neuronal. También se deben calcular las derivadas del sesgo, que indican en qué capa se encuentra el error. El uso de estas derivadas parciales permite encontrar el error, y lo que realiza el algoritmo es retroceder hasta la neurona donde se encuentra el error, regresando desde la capa de salida hasta la capa de entrada.

Para poder realizar todo este proceso de retropropagación, es necesario contar con una función de activación diferenciable.

8.5. Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales (CNN, por sus siglas en inglés) son redes profundas con una estructura especial. Están conformadas por tres tipos de capas: convolucionales, de agrupación y completamente conectadas.

En las capas convolucionales, el filtro detecta características específicas dentro de la imagen de entrada, como bordes, colores, o texturas, y genera un mapa de características. En las capas de agrupación, se reduce la resolución espacial de la imagen, lo que permite reducir la cantidad de parámetros y el costo computacional. Finalmente, las capas completamente conectadas están encargadas de realizar la clasificación.

Las CNN son muy útiles para el reconocimiento de imágenes, y se han utilizado en una amplia variedad de aplicaciones, desde la visión por computadora hasta la medicina y el reconocimiento de voz.

8.6. Aprendizaje Incremental

Con el paso del tiempo, la tecnología ha evolucionado, lo que ha llevado a un aumento en la cantidad de datos disponibles. El aprendizaje automático ha experimentado avances significativos, y los datos se generan y procesan con mayor frecuencia.

Se puede definir una tarea de aprendizaje como incremental si los ejemplos de entrenamiento utilizados para resolverla se presentan de manera secuencial, generalmente uno a la vez. Si los resultados no son urgentes, este tipo de tareas puede ser resuelto mediante algoritmos de aprendizaje no incremental [5]. Un área donde el aprendizaje incremental es especialmente útil es en la *robótica*, ya que estos sistemas requieren entrenamiento constante [5].

Este enfoque de aprendizaje fue inspirado por la forma en que los seres humanos aprenden y es más rápido, lo que llevó a su adopción en el campo del aprendizaje automático.

Con el tiempo, el aprendizaje incremental se ha convertido en un paradigma del aprendizaje automático, donde el sistema toma nuevos ejemplos y los agrega a los ya aprendidos. A medida que el sistema aprende, los ejemplos previos pueden ser reemplazados por los nuevos [7].

8.6.1. Algoritmos de Aprendizaje Incremental

Un algoritmo de aprendizaje incremental se define por los siguientes criterios:

1. Ser capaz de aprender y actualizarse con cada nuevo dato, etiquetado o no etiquetado.
2. Conservar el conocimiento adquirido previamente.
3. No requerir acceso a los datos originales.
4. Ser capaz de generar nuevas clases o clusters cuando sea necesario, así como dividir o fusionar clusters según lo requiera el entorno.
5. Tener una naturaleza dinámica, adaptándose a un entorno cambiante [1].

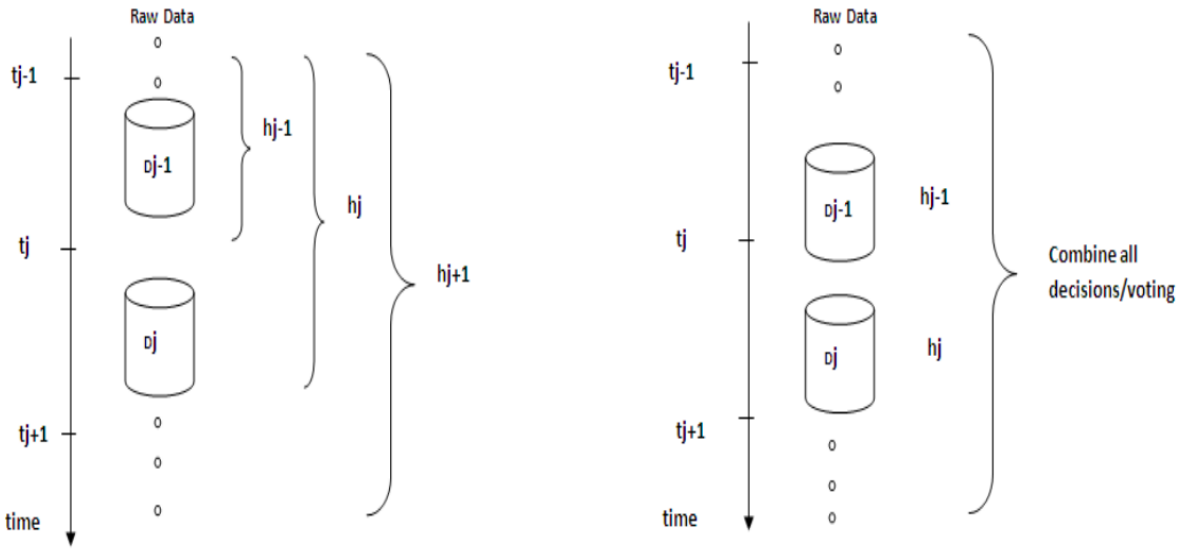


Figura 11: Dos enfoques tradicionales del aprendizaje incremental.

Como se observa en la Figura 11, el primer enfoque consiste en la acumulación de datos, donde, al recibir una nueva porción de datos D_j, se descarta la hipótesis h_{j-1} y se genera una nueva hipótesis h_j basada en todos los datos disponibles hasta ese momento. En el segundo enfoque, al recibir una nueva porción de datos D_j, se desarrolla una única hipótesis nueva o un conjunto de hipótesis nuevas basadas en los nuevos datos. Finalmente, se puede usar un mecanismo de votación para combinar todas las decisiones de las diferentes hipótesis y obtener la predicción final.

El aprendizaje incremental tiene la ventaja de no requerir almacenamiento de los datos previos, ya que el conocimiento se guarda en las hipótesis generadas durante el proceso de aprendizaje.

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

$$e_1, e_2, ..., e_s$$

produce una secuencia de hipótesis

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

tal que h_{i+1} depende solo de h_i y de la muestra actual e[5]."

Como se observa, estos algoritmos permiten que la inteligencia artificial realice tareas de predicción de manera más eficiente.

Un ejemplo de esta metodología es el proyecto *COBWEB*, que categoriza el número de clusters y la pertenencia de dichos clusters utilizando una métrica probabilística global. Este proceso implica agregar nuevas categorías, actualizando las probabilidades con los nuevos datos recolectados [4].

8.7. Estado del Arte

9. Metodología

Este trabajo se divide en tres etapas principales: la recreación del modelo base, la implementación de una extensión al modelo, y la comparación de los resultados obtenidos. A continuación, se describen cada una de estas etapas en detalle:

1. **Recreación del código base**
   Como primer paso, se recreará el código presentado en [2], que describe la implementación de una red neuronal multicapa utilizando el algoritmo de entrenamiento *backpropagation* en el lenguaje de programación Python. Este código servirá como base para las siguientes etapas.
   Además, se empleará el conjunto de datos *Optical Digits*, el cual será sometido a un proceso de preprocesamiento para eliminar registros inválidos y garantizar la calidad de los datos utilizados en el entrenamiento y validación del modelo.

2. **Extensión del modelo base**
   Posteriormente, se desarrollará una extensión al modelo base, con el objetivo de experimentar con configuraciones más complejas de la red neuronal. Específicamente, se agregarán más de dos capas de pesos duplicados, lo cual se espera que mejore la retención de información al emplear el aprendizaje incremental.
   Para este propósito, se realizarán experimentos incrementando gradualmente el número de capas de pesos duplicados, con el fin de analizar el impacto de esta configuración en la capacidad de aprendizaje de la red.

3. **Comparación y análisis de resultados**
   Una vez obtenidos los resultados de las simulaciones, se realizará una comparación entre el rendimiento del modelo base y el modelo extendido. Este análisis incluirá métricas clave, como la precisión del modelo, la tasa de olvido de información y el desempeño en el aprendizaje incremental.
   Los resultados permitirán evaluar si las modificaciones introducidas en la arquitectura de la red neuronal mejoran significativamente su desempeño en comparación con la implementación original.

10. Organización del Capitulado

En el capitulo 2 se explicará lo que son y como funcionan las redes neuronales artificiales, así como 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 el conjunto de datos Optdigit que se utilizará para el entrenamiento y prueba de los algoritmos. 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.

11. Diagrama de Gantt

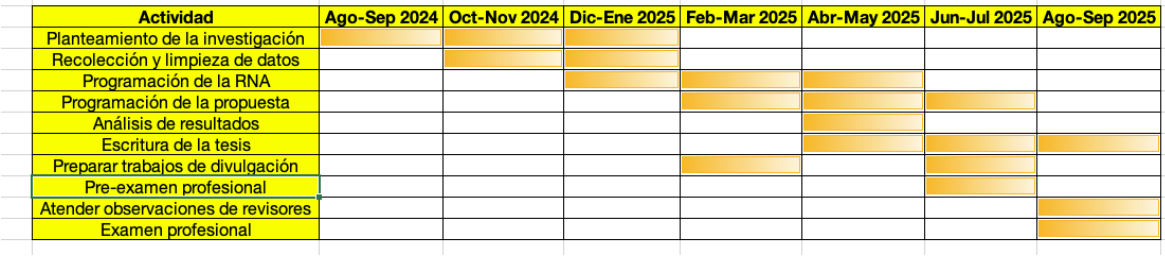


Figura 12: Diagrama de Gantt

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

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

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

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

[6] 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.

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

[8] V Renganathan. Overview of artificial neural network models in the biomedical domain. *Bratislava Medical Journal/Bratislavské Lekárske Listy*, 120(7), 2019.

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

[10] 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.

[11] 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.