UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ciencias y Humanidades

Teorema de Aproximación Universal

Trabajo de graduación en modalidad de Tesis presentado por
José Eduardo López Gómez
para optar al grado académico de Licenciado en Matemática Aplicada

Guatemala,

2023

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ciencias y Humanidades

Teorema de Aproximación Universal

Trabajo de graduación en modalidad de Tesis presentado por
José Eduardo López Gómez
para optar al grado académico de Licenciado en Matemática Aplicada

Guatemala,

2023

Vo.Bo.:

(f) _____

MSc. Paulo Mejía Castillo

Tribunal Examinador:

(f) _____

MSc. Paulo Mejía Castillo

(f) _____

MA. Dorval Carías Samayoa

(f) _____

MSc. Alan Reyes

Fecha de aprobación: Guatemala, 12 de junio de 2023.

Prefacio

En el presente trabajo de tesis, se aborda un tema fascinante en el campo de las Redes Neuronales: el teorema de aproximación universal. Este teorema es fundamental en el estudio de las capacidades de las Redes Neuronales para aproximar cualquier función continua, lo que las convierte en herramientas poderosas para el procesamiento de información. Además, se explora una aplicación específica de este teorema: el desarrollo de un modelo capaz de interpretar el alfabeto en lenguaje de señas en tiempo real.

El Teorema de Aproximación Universal ha sido objeto de intensa investigación en los últimos años. Este teorema demuestra que una Red Neuronal con una arquitectura adecuada puede aproximar cualquier función continua en un dominio dado. El estudio de este teorema nos permite comprender las capacidades y limitaciones de las Redes Neuronales en términos de su representación y adaptabilidad a diferentes tipos de problemas.

Además, se presenta un prototipo que utiliza la teoría del Teorema de Aproximación Universal para interpretar el alfabeto en lenguaje de señas en tiempo real. El objetivo es proporcionar una aplicación práctica del teorema de Aproximación Universal, que tenga la capacidad de reconocer y traducir el alfabeto de lenguaje de señas guatemalteco de forma efectiva y fluida, utilizando técnicas de visión computacional y el uso de una Red Neuronal entrenada.

Agradecimientos

A mi familia y profesores, en especial a mi madre, y mis abuelos, que en paz descansen.

Índice

Lista de figuras

Lista de cuadros

Resumen

Este trabajo se enfoca en dos importantes contribuciones al campo de las Redes Neuronales y sus fundamentos teóricos. Por un lado, se analiza en profundidad el Teorema de Aproximación Universal de Cybenko y su demostración. La demostración de este teorema utiliza herramientas de ramas como Teoría de la Medida y el Análisis Funcional, más específicamente, el Teorema de Hahn - Banach, el Teorema de Representación de Riesz y el Teorema de Convergencia Dominada de Lebesgue. No obstante, se requiere un conocimiento básico de Análisis de Variable Real, Topología y Teoría de Conjuntos. Asimismo, se presente una breve historia de cómo surge el interés de investigación y aplicación de las Redes Neuronales, sus orígenes y contexto histórico.

Por otro lado, se presenta la versión extendida del teorema de Cybenko propuesta por Hornik, White y Stinchcombe. Esta versión extiende el teorema original, ya que permite utilizar funciones de activación más generales y la extensión de $I_n = [0, 1]^n$, a un conjunto compacto general. Para la demostración de este teorema, se utiliza principalmente el teorema Stone-Weierstrass y bastantes herramientas de Análisis de Variable Real. Esto proporciona los fundamentos matemáticos necesarios para llevar a cabo la demostración en Redes Neuronales de una sola capa oculta con una función de activación continua y no constante de un espacio de funciones continuas sobre un compacto.

Finalmente, se implementó un prototipo de aplicación de los resultados anteriormente mencionados, esto con la finalidad de mostrar la utilidad que tienen dichos algoritmos pese a ser las estructuras de Red Neuronal más básicas. Para ello, se desarrolló dos modelos en el lenguaje de programación Python, con el apoyo de diversas librerías, como OpenCV, MediaPipe de Google y Keras, para interpretar el alfabeto de LENSEGUA (lenguaje de señas guatemalteco oficial) en tiempo real. Esto implicó, la recopilación de los datos, el procesamiento de imágenes utilizando algoritmos de visión computacional, el entrenamiento de los modelos y la implementación de el programa capaz de ejecutarlos en tiempo real.

CAPÍTULO 1

Introducción

Dentro de la rama del aprendizaje automático, abunda una vasta cantidad de aplicaciones prácticas. Sin embargo, a pesar de que la teoría que fundamenta estas aplicaciones es extensa en raras ocasiones se presenta o habla de ella, incluso, hay teoremas que destacan por su profundo impacto en la comprensión y aplicación de conceptos fundamentales. Entre ellos, se encuentran el Teorema de Aproximación Universal propuesto por Cybenko y su posterior extensión demostrada por Hornik, White y Stinchcombe. Estos teoremas constituyen un hito en el estudio de las Redes Neuronales y su capacidad para aproximar funciones continuas.

Además, es necesario resaltar que las Redes Neuronales han ganado una gran relevancia en los últimos años debido a su capacidad para modelar y resolver problemas complejos en diversos campos, como la Inteligencia Artificial, el procesamiento de señales y el aprendizaje automático. En este contexto, el Teorema de Aproximación Universal ha despertado un gran interés, ya que establece que una Red Neuronal con una sola capa oculta puede aproximar cualquier función continua.

Este trabajo tiene como finalidad presentar la teoría que fundamenta estos teoremas, sus demostraciones y sumergir al lector en la relevancia que generan las Redes Neuronales desde una perspectiva teórica. Por ello, se explora a fondo el Teorema de Aproximación Universal de Cybenko, en conjunto con sus limitantes. Además, una vez vista esta versión, se presenta la versión extendida de Hornik, Stinchcombe y White, junto a la teoría matemática que lo fundamenta. Si bien, existen varias versiones del Teorema de Aproximación Universal este trabajo se centra principalmente en las versiones que abarcan las Redes Neuronales de Propagación hacia Adelante de una sola capa oculta. Cabe resaltar, que la teoría y demostraciones presentadas a lo largo de este documento se realizaron con un alto nivel de detalle para facilitar su comprensión. No obstante, se requiere conocimientos previos de Análisis de Variable Real, Cálculo, Análisis Funcional, Teoría de la Medida, Teoría de Conjuntos y Topología para comprender las demostraciones en su totalidad.

Por último, se trabajó una aplicación práctica del teorema, la cual se basa en construir un prototipo capaz de identificar el alfabeto de LENSEGUA en tiempo real. Para ello, se hizo uso del lenguaje de programación Python, si bien en el documento no se presenta el código de los programas utilizados, si se presentan los modelos de redes generados, las librerías empleadas durante la recopilación y el procesamiento de los datos, el entrenamiento de los modelos y su funcionamiento en tiempo real. Cabe resaltar que esta sección no requiere un conocimiento vasto de Matemática, dado que es una descripción de cada una de las etapas del desarrollo del prototipo.

CAPÍTULO 2

Objetivos

2.1. Objetivo general

Presentar la teoría que fundamenta el Teorema de Aproximación Universal, su demostración y aplicación a un modelo.

2.2. Objetivos específicos

- Mostrar la teoría matemática requerida para llevar a cabo la demostración.
- Demostrar de forma detallada el Teorema de Aproximación Universal.
- Presentar una aplicación práctica del teorema con su modelo respectivo, en este caso se trabajará con una aplicación de reconocimiento del alfabeto en lenguaje de señas básico para el contexto de Guatemala.

CAPÍTULO 3

Justificación

Durante la última década, los algoritmos de aprendizaje automático y aprendizaje profundo se han convertido en pioneros de avances tecnológicos y desarrollo de nuevas herramientas que permiten la automatización de procesos. Por esta razón, es importante conocer de forma estructurada su funcionamiento y el límite que dichos algoritmos pueden tener. De esta forma, se puede considerar al Teorema de Aproximación Universal cómo uno de los fundamentos más importantes de la teoría de Redes Neuronales.

Hoy en día se ha popularizado mucho el uso de Redes Neuronales para resolver problemas prácticos, dado que son bastante versátiles y pueden acoplarse fácilmente a las necesidades del usuario. No obstante, nos encontramos en una época donde reina lo empírico, y se ha dejado de lado las teorías matemáticas que fundamentan el porqué de las incógnitas que nos planteamos día con día. Por esta razón, es necesario dar a conocer la demostración de este teorema y recalcar la importancia de sus aplicaciones tanto teóricas cómo prácticas.

CAPÍTULO 4

Historia de las Redes Neuronales

Las redes neuronales comenzaron siendo un concepto estudiado en 1943 por el neurofisiólogo Warren McCulloch y el matemático Walter Pitts, quienes escribieron un artículo que explicaba la hipótesis acerca del funcionamiento de las neuronas en el cerebro humano. Para explicar la transmisión de información entre neuronas modelaron una pequeña red neuronal como un circuito eléctrico. Posteriormente, en el año 1949, Donal Hebb escribió *The Organization of Behavior*, una investigación que se enfocó en señalar que las conexiones neuronales se fortalecen cada vez que se utilizan, un concepto fundamental para las forma de aprendizaje de los seres humanos. La razón de este argumento es que al momento en que dos neuronas envían una señal entre ellas estas refuerzan la conexión que las une. (Clabaugh, Myzewski, y Pang, 2023) (Masher y D'Bannon, 2016)

No obstante, no fue hasta la mitad del siglo XX durante la época de los años 1950, las computadoras avanzaron de una forma significativa, lo cual permitió la simulación hipotética de una Red Neuronal. Los laboratorios de investigación de IBM fueron los primeros en adentrarse en hacer este tipo de pruebas, lamentablemente, los resultados no fueron los esperados. Casi una década más tarde dos estudiantes de la Universidad de Stanford, Bernard Widrow y Marcian Hoff, desarrollaron dos modelos a los que nombraron *ADALINE* y *MADALINE*, los cuales se refieren a *ADAptive Linear Elements* y *Multiple ADAptive Linear Elements* respectivamente. El primero fue diseñado para reconocer patrones binarios con la finalidad de leer cadenas de bits de una línea telefónica y que fuera capaz de predecir el siguiente bit. En el caso de *MADALINE*, esta fue la primera Red Neuronal empleada para tratar un problema de la vida real. Haciendo uso de un filtro adaptativo, el modelo consiguió eliminar, en su mayoría, el eco de las líneas telefónicas. A pesar de que el sistema era bastante rudimentario, llegó a tener un uso comercial. Cabe recalcar que estos modelos no fueron las únicas contribuciones de estos investigadores, en el año 1962, idearon un procedimiento de aprendizaje capaz de examinar los valores para ajustar las entradas de una red neuronal (0 y 1 en este caso), según su método se puede estimar el cambio de peso con la siguiente fórmula:

$$WC = \frac{PW * \epsilon}{\#inputs}$$

donde WC se refiere al cambio de cada peso, PW al valor previo al peso, ϵ al error y los *inputs* se refiere a las cadenas binarias compuestas por 0's y 1's. La fórmula se rige bajo la premisa de que si alguna neurona percibe un error bastante alto, entonces es posible ajustar los pesos para propagar o mitigar el error a lo largo de la red, o al menos, sobre las neuronas cercanas. (Clabaugh y cols., 2023) (Masher y D'Bannon, 2016)

A pesar del éxito de las redes neuronales, la tradicional arquitectura de von Neumann captó mayor

atención sobre la computación de la década de los sesenta. Esto conllevó a que las investigaciones y el estudio de las redes neuronales se dejara atrás. Durante esta época se publicó una investigación que estimaba que no era posible expandir las Redes Neuronales más allá de una sola capa a tener una red más compleja con múltiples capas. Además, la mayor cantidad de personas vinculadas a la investigación de las redes se caracterizaba por el uso de una función no diferenciable sobre todo la recta real. Como consecuencia, la mayoría de fondos destinados al financiamiento de este campo bajó considerablemente, por lo que de cierta forma el crecimiento de estos algoritmos se vio estancado durante varios años. Esto se argumentó con el hecho que el éxito prematuro de las redes en sus primeros años llevaran a una exageración de su potencial, especialmente cuando se consideran las limitaciones tecnológicas de la época. Las metas trazadas a lo largo de los años para las redes neuronales no fueron alcanzadas, y de la mano con una implementación bastante compleja sumado a la gran popularidad de las estructuras de von Neumann generaron pequeños avances pero no tan significativos como los de hoy en día. Por otro lado, fue hasta 1972 cuando Kohoren y Anderson desarrollaron un red similar e mutuamente independiente. Al implementar matrices sobre el modelo de Widrow y Marcian se logró crear vectores de circuitos análogos de modelos *ADALINE*, esto comenzó con lo que hoy se conoce como un conjunto de funciones de activación, los cuales eran capaces de tener más de una neurona de salida, y posteriormente en 1975, se desarrollo la red de capas múltiples. (Clabaugh y cols., 2023)

Finalmente, llegamos a los años ochenta, una etapa que renovó el interés sobre la investigación relacionadas a las Redes Neuronales. A comienzos de la década, en el año 1982, John Hopfield del Caltech presentó un artículo a la Academia Nacional de Ciencias, su idea se basó en crear máquinas más útiles empleando vías bidireccionales entre neuronas. Durante ese mismo año Reilly y Cooper presentaron una «red híbrida» con múltiples capas, cada capa utilizaba una función diferente. En adición a los avances tecnológicos también hubo una razón política para incrementar el financiamiento dedicado al desarrollo de Redes Neuronales. Durante una conferencia entre Estados Unidos y Japón sobre Redes Neuronales Competitivas y Cooperativas el país asiático anunció su interés y esfuerzo por producir una quinta generación de redes, lo que generó cierta inconformidad en los investigadores estadounidenses quienes temían ser rebasados en el campo. Para fines prácticos, en este ámbito se conoce como quinta generación de Redes Neuronales a la que contempla e involucra la Inteligencia Artificial, de forma más concreta se listan las cinco generaciones a continuación:

- **Primera generación**: generación de interruptores y cables, la más básica y rudimentaria.
- **Segunda generación**: implementó transistores con la intención de mejorar la primera.
- **Tercera generación**: comenzó a implementar circuitos más complejos y lenguajes de programación.
- **Cuarta generación**: se basó completamente en código, buscaba crear generadores de código.
- **Quinta generación**: implementación de la Inteligencia Artificial.

Esto tuvo repercusiones bastante importantes en los avances tecnológicos de los algoritmos de redes, en 1986 por ejemplo, las Redes Neuronales de capas múltiples eran la tendencia de ese entonces y muchos investigadores estaban intentando extender el modelo de Widrow - Hoff para dichas redes. Entonces, un miembro del departamento de psicología de Stanford, David Rumelhart, ideó lo que hoy en día se conoce como redes de *back propagation*. Estas se caracterizan por sus patrones de reconocimiento de errores a lo largo de todo el circuito de la red. A diferencia de las redes híbridas que implementaban dos capas, las redes de de Rumelhart podían implementar una gran cantidad de capas. A pesar de que esto suena como una ventaja inmensa sobre los algoritmos anteriormente diseñados, las redes que implementan el *back propagation* tiene una desventaja bastante grande. Se les denomina redes de aprendizaje lento debido a que necesitan una gran cantidad de iteraciones para poder aprender. (Clabaugh y cols., 2023)

Una vez asentadas las bases de las Redes Neuronales, los matemáticos se dedicaron a una exhaustiva investigación sobre estos algoritmos, lo que llevó a lo que se conoce hoy en día como el Teorema Universal de Aproximación o Teorema de Aproximación Universal. Este teorema tiene varias versiones, cada una específica para un distinto tipo de red, la versión mas simple fue demostrada en el año 1989 por George Cybenko (College, 2023), la cual trabaja sobre una red de una sola capa densa bajo la premisa del uso de funciones sigmoide como funciones de activación. Ese mismo año, tres investigadores de la Universidad de San Diego, Kurt Hornik, Maxwell Stinchcombe y Halbert White presentaron una demostración para Redes Neuronales de múltiples capas. Desde entonces se han presentado una gran cantidad de demostraciones para el teorema acorde a características específicas de una red. No obstante, hasta 2019 se demostró el caso para Redes Neuronales Convolucionales (CNN por sus siglas en inglés), esto fue de gran importancia dado que en la última década se ha popularizado el uso de visión computacional para tratar problemas modernos. Un claro ejemplo de ello es el reconocimiento facial o el reconocimiento de objetos que emplean los vehículos autónomos, el autor de la demostración fue un Profesor de la Escuela de Ciencias de Datos de la Universidad de Hong Kong, Ding-Xuan Zhou. (Sciences, 2023)

CAPÍTULO 5

Introducción a las Redes Neuronales

5.1. ¿Qué es una red neuronal?

Una red neuronal es un modelo matemático que tiene la finalidad de reconocer patrones o relaciones en un conjunto de datos mediante la mimetización del proceso que opera dentro del cerebro humano. En este sentido, una red neuronal se refiere a un sistema de neuronas, sin importar si son orgánicas o artificiales. Más a fondo, una neurona dentro de este sistema es una función matemática que se encarga de la recolección y clasificación de información mediante una arquitectura específica, por ende, tienen un gran parecido con los métodos estadísticos como el ajuste de curvas y las regresiones lineales o múltiples. Las neuronas dentro de las Redes Neuronales pueden ser organizadas mediante capas interconectadas por nodos, a cada uno de estos nodos se les conoce como perceptrón, los cuales se encargan de alimentar una función de activación (Chen, 2023), (Calin, 2020). Por lo general, una red neuronal se constituye de tres tipos de capas:

- **Capa de entrada**: esta corresponde a las variables a ingresar a la red, generalmente proviene de los datos obtenidos. Esta es una variable que se introduce al sistema, y puede ser unidimensional $x \in \mathbb{R}$, vectorial $\hat{x} \in \mathbb{R}^n$, una matriz $x \in \mathbb{M}^{n\times n}$ o incluso una variable aleatoria X (Chen, 2023), (Calin, 2020).
- **Capa oculta**: esta capa contiene los nodos y es donde se encuentran las funciones de activación. Los nodos o neuronas utilizan estas funciones de activación para modificar las variables de ingreso de la red. Por lo general, las redes neuronales más sofisticadas cuentan con múltiples capas ocultas, cada una con una cantidad de nodos independientes en cada capa. En este punto la red actúa como una función, que modifica las variables de la capa de entrada de cierta forma para producir un resultado de salida. La forma en que estas entradas se modifican para obtener la variable de salida es mediante dos parámetros, los pesos y los sesgos denotados por (w, b) los cuales pueden ser unidimensionales, un vector, o una variable aleatoria (Chen, 2023), (Calin, 2020).
- **Capa de salida**: dentro de esta capa es donde se obtiene los resultados de la red neuronal, esta puede ser unidimensional $y \in \mathbb{R}$, un vector $\hat{y} \in \mathbb{R}^n$, una variable aleatoria Y o un tensor (Chen, 2023), (Calin, 2020).

Dicho esto podemos pensar en una red neuronal de la siguientes formas: $y = f_{w,b}(x)$ para el caso univariable, $\hat{y} = f_{w,b}(\hat{x})$, $Y = f_{w,b}(X)$ para el caso de variables aleatorias. No obstante, de esto pueden ocurrir mezclas tal como tener una variable de entrada multidimensional y una variable de salida unidimensional $y = f_{w,b}(\hat{x})$ por ejemplo. Además, podemos considerar la red neuronal más simple como aquella que se compone de una capa de entrada, una capa oculta con un solo nodo que cuenta con una función de activación y una capa de salida (Calin, 2020). Recordemos que estos son los componentes mínimos requeridos para trabajar con una red, sin embargo, existen distintos tipos de redes neuronales. No obstante, en este trabajo se hará énfasis en las redes neuronales de propagación hacia adelante de una sola capa (**sección 8.1**) y las redes neuronales densas de una sola capa (**sección 8.2**).

5.2. Tipos de funciones de activación

Las redes neuronales necesitan de funciones de activación para poder aprender. Sin embargo, la mayoría de veces estas funciones de activación no son lineales, y esto modifica directamente el funcionamiento de la red. De tal forma, el hecho de escoger una función de activación puede cambiar el comportamiento de una red por completo (Calin, 2020). Dentro del mundo de las funciones de activación puedes encontrar cuatro tipos esenciales:

- Funciones lineales
- Funciones escalón
- Funciones Palo de Hockey
- Funciones Sigmoide

Cada una de las funciones anteriormente mencionadas tiene sus ventajas y desventajas en cuanto a ser continuas, diferenciales, acotadas, etc.

5.2.1. Funciones lineales

La pendiente de una recta puede utilizarse para modelar la razón de salida de una neurona. Una función de activación lineal se emplea mayormente en redes neuronales multicapa para la capa de salida. Además, es importante mencionar que incluir una función de activación lineal es equivalente a realizar una regresión lineal. Dicho esto, si se considere el caso donde $f(x) = kx$ y $k > 0$ es una constante positiva, entonces se tiene que la razón de salida es la derivada $f'(x) = k$ por lo que es constante. En particular, si se considera el caso $k = 1$, se tiene la función de activación identidad, y la razón de salida es $f'(x) = 1$ (Calin, 2020).

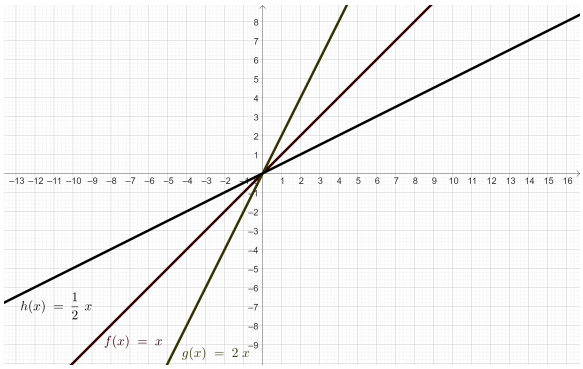

Figura 5.1: Gráfico de distintas funciones lineales.

5.2.2. Funciones escalón

Este tipo de funciones se caracteriza por tener un salto en algún punto de su dominio. Por ende, no son derivables en dicho punto y tampoco son continuas. Sin embargo, son bastante útiles cuando queremos enfocarnos en una razón de salida que solo acepte valores mayores a un cierto criterio (Calin, 2020). Por ejemplo, tenemos la función escalón unitario o también conocida como función Heaviside definida por:

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

en este caso particular la función nos ayuda a filtrar los valores positivos $x \geq 0$. Como se mencionó anteriormente, este tipo de funciones no son diferenciales en un punto específico, en este caso $x = 0$. Sin embargo, cuando hablamos en términos generales (Calin, 2020), la derivada de esta función está dada por la función delta de Dirac, $H'(x) = \delta(x)$ donde

$$\delta(x) = \begin{cases} 0, & x \neq 0 \\ \infty, & x = 0 \end{cases}$$

Otro ejemplo es la función signo, y se define de la siguiente manera:

$$S(x) = \begin{cases} -1, & x < 0 \\ 1, & x \geq 0 \end{cases}$$

Lo interesante de esta función es que puede relacionarse con la función Heaviside de la siguiente forma (Calin, 2020):

$$S(x) = 2H(x) - 1$$

y por ende su derivada también está definida por la función delta de Dirac:

$$S'(x) = 2H'(x) = 2\delta(x)$$

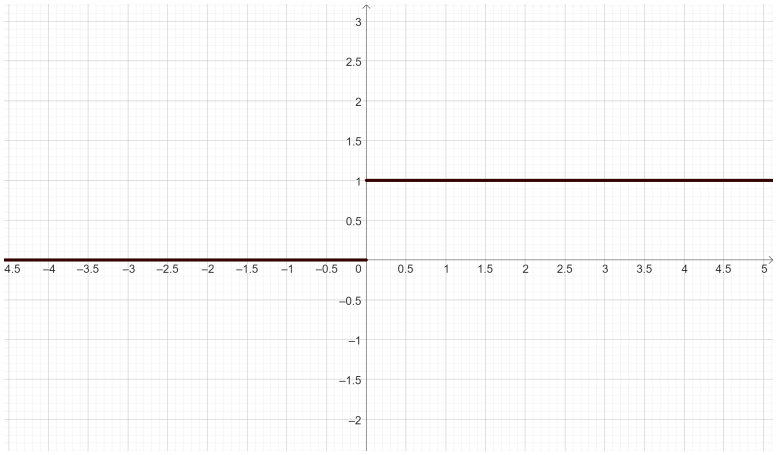

Figura 5.2: Gráfico de la función Heaviside.

Figura 5.3: Gráfico de la función signo.

5.2.3. Funciones Palo de Hockey

Estas funciones reciben su nombre debido a que su gráfico es bastante similar a un palo de hockey, tienen una forma de L curvada (Calin, 2020). Una de las más conocidas es la función ReLU, por sus siglas en inglés que significan *Rectified Linear Unit* y puede definirse de las siguientes tres formas:

$$ReLU(x) = xH(x) = \text{máx}\{x, 0\} = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}$$

Además, podemos observar que la derivada de esta función es la función Heaviside, por lo que:

$$ReLU'(x) = H(x)$$

Esta es una función bastante útil debido a que en problemas que requieren demasiado poder de procesamiento hace que la red aprende de forma más rápida en comparación con otras funciones de activación (Calin, 2020). Sin embargo, no deja de ser la función más simple de palo de hockey, y por ende se han generado versiones más versátiles como la función PReLU (*Parametric Rectified Linear*

Unit) que se define de la siguiente manera:

$$PReLU(\alpha, x) = \begin{cases} \alpha x, & x < 0 \qquad \alpha > 0 \\ x, & x \geq 0, \end{cases}$$

A diferencia de la función ReLU, PReLU por lo general no filtra los valores negativos. No obstante, podemos observar que tenemos las siguientes equivalencias para valores específicos de α:

$$PReLU(0, x) = ReLU(x), \qquad PReLU(1, x) = f(x) = x$$

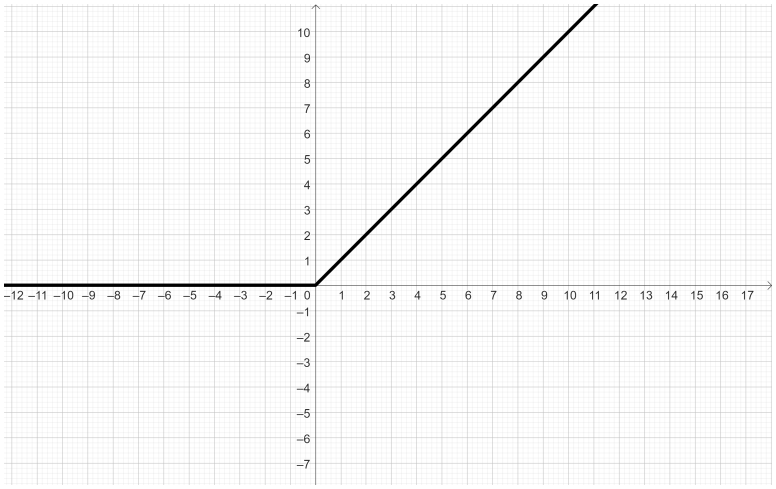

Figura 5.4: Gráfico de la función ReLU.

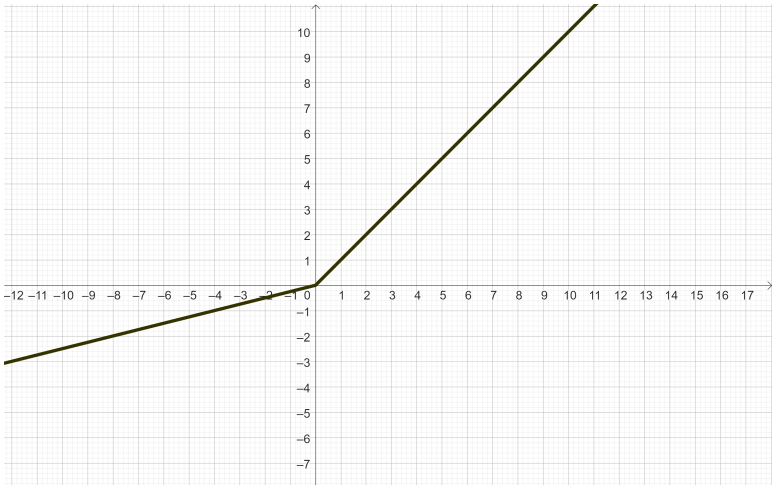

Figura 5.5: Gráfico de la función PReLU.

Dentro de esta categoría de funciones también podemos encontrar algunas que trabajan con funciones exponenciales, de estas la más simple es la ELU (*Exponential Linear Unit*) definida por:

$$ELU(\alpha, x) = \begin{cases} \alpha(e^x - 1), & x \leq 0, \qquad \alpha > 0 \\ x, & x > 0 \end{cases}$$

En este caso, se debe notar que la función es diferenciable en $x = 0$ únicamente cuando $\alpha = 1$. En caso contrario, la función genera una discontinuidad en su derivada (Calin, 2020). Sin embargo, se puede pensar en un caso más general de la función ELU, para ello tenemos la función SELU (*Scaled Exponential Linear Unit*):

$$SELU(\alpha, \lambda, x) = \begin{cases} \lambda\alpha(e^x - 1), & x \leq 0, \qquad \alpha > 0, \lambda > 0 \\ \lambda x, & x > 0, \qquad \lambda > 0 \end{cases}$$

Cuando se agrega el parámetro λ entonces se puede definir la función de tal forma que ésta sea diferenciable siempre y cuando $\alpha = 1$ (Calin, 2020).

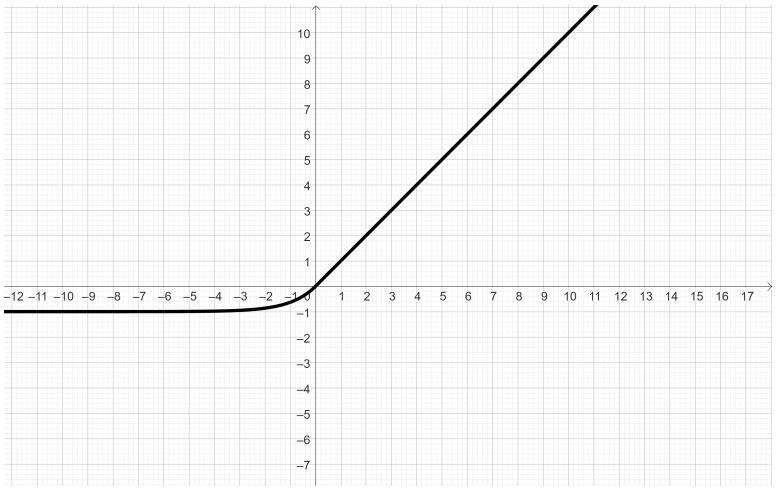

Figura 5.6: Gráfico de la función ELU.

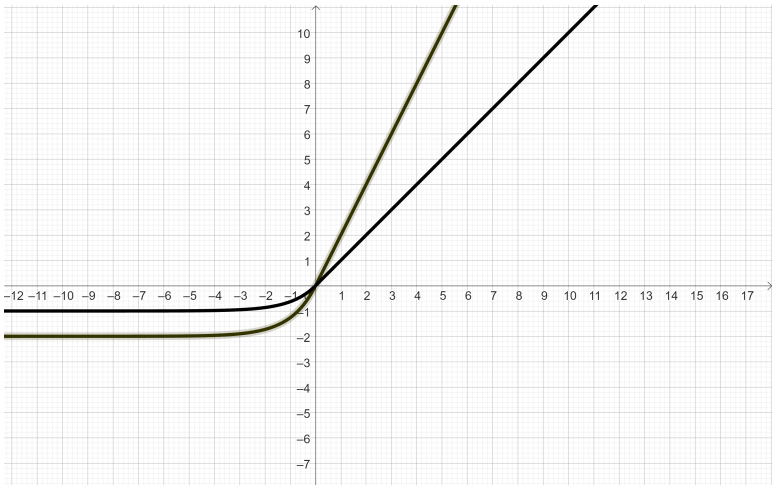

Figura 5.7: Gráfico de distintas funciones SELU.

5.2.4. Funciones Sigmoide

Este tipo de funciones tiene la ventaja de ser analíticas y pueden aproximar una función escalón con una precisión bastante alta. Además, cuando el rango de estas funciones se encuentra entre el intervalo $[0, 1]$ los resultados pueden ser interpretados como una probabilidad (Calin, 2020). Estas

funciones tienen la peculiaridad de tener dos asíntotas horizontales en $y = 0$ y $y = 1$. Más adelante, en el capítulo 6, se definirá formalmente lo que es una función sigmoide en el sentido matemático, por el momento se citan algunos ejemplos como la función logística:

$$\sigma(c, x) = \frac{1}{1 + e^{-cx}}, \qquad c > 0$$

Esta función también es conoce como la función escalón unitario suave (Calin, 2020). Esto porque cuando $c \to \infty$ la función $\sigma(c, x)$ aproxima bastante bien a la función Heaviside $H(x)$, pero con la propiedad de ser analítica:

$$\lim_{c\to\infty} \frac{1}{1 + e^{-cx}} = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} = H(x), \qquad x \neq 0$$

No obstante, el punto $x = 0$ tiene una peculiaridad, y es que en la función logística converge a $y = 1/2$, por ello se dice que es una buena aproximación, más no es exactamente la función Heaviside en todo su dominio. Además, es importante notar que en este caso el parámetro c es el encargado de controlar la velocidad de ajuste de los pesos. Al calcular la derivada de esta función:

$$\sigma'(c, x) = \frac{ce^{-cx}}{(1 + e^{-cx})^2} = c\left[\frac{1 + e^{-cx} - 1}{(1 + e^{-cx})^2}\right] = c\left[\frac{1}{1 + e^{-cx}} - \frac{1}{(1 + e^{-cx})^2}\right] = c(\sigma(c, x) - \sigma^2(c, x))$$

y por lo tanto:

$$\sigma'(c, x) = c\sigma(c, x)(1 - \sigma(c, x))$$

Más importante aún, cuando el parámetro $c = 1$ entonces se obtiene lo que se conoce como la función logística estándar y se obtiene:

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

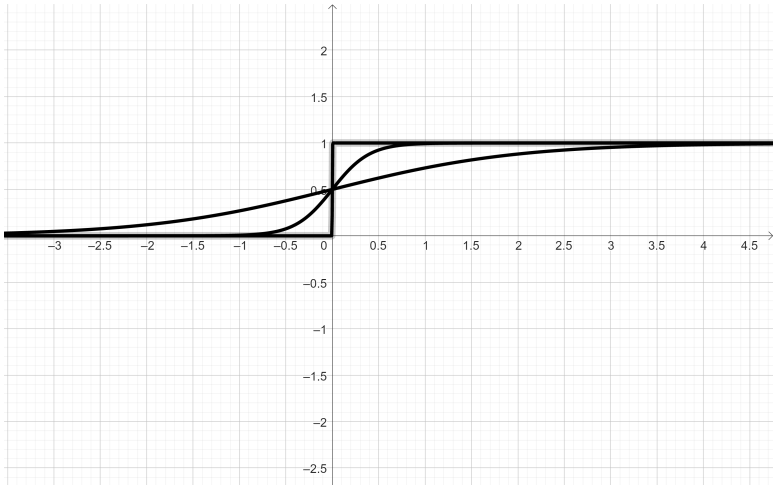

Figura 5.8: Gráfico de distintas funciones sigmoide aproximando a la función Heaviside.

Otro ejemplo es la función tangente hiperbólica, que también es conocida como la función sigmoide bipolar y se define de la siguiente manera:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{1 - e^{-2x}}{1 + e^{-2x}} = \frac{2 - e^{-2x} - 1}{1 + e^{-2x}} = \frac{2}{1 + e^{-2x}} - 1 = 2\sigma(2, x) - 1$$

se puede observar que al calcular su derivada se obtiene lo siguiente:

$$\tanh'(x) = 4\sigma(2, x)(1 - \sigma(2, x)) = 2\sigma(2, x)(2 - 2\sigma(2, x)) = (1 + \tanh(x))(1 - \tanh(x)) = 1 - \tanh^2(x)$$

esta función está centrada en el origen, lo cual representa una ventaja en comparación a la función $\sigma(x)$.

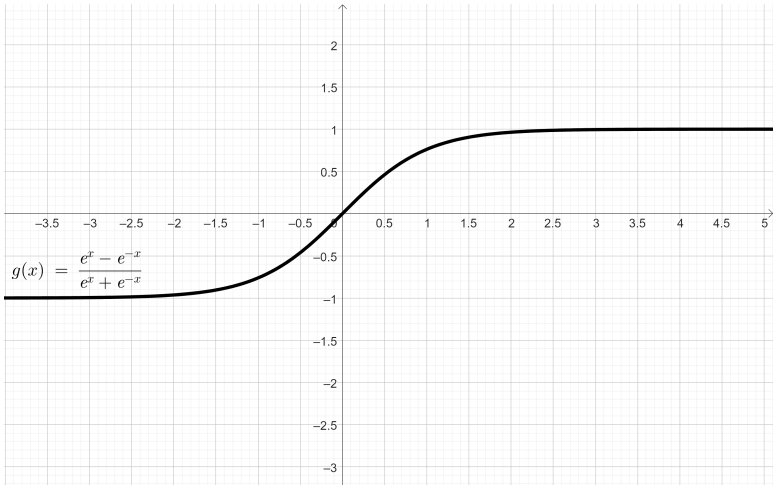

Figura 5.9: Gráfico de la función $\tanh(x)$

Un ejemplo más de una función sigmoide es la función arcotangente multiplicada por una constante específica:

$$h(x) = \frac{2}{\pi} \arctan(x)$$

la cual es analítica y su derivada está dada por:

$$h'(x) = \frac{2}{\pi(1 + x^2)}$$

La función $h'(x)$ alcanza su máximo en $x = 0$ y cuando $x \to -\infty$ o $x \to \infty$ entonces $h'(x) \to 0$, por lo que extrae mayor información de valores de x pequeños, a diferencia de los grandes que no presentan mayor relevancia en cuanto a la modificación de los pesos (Calin, 2020).

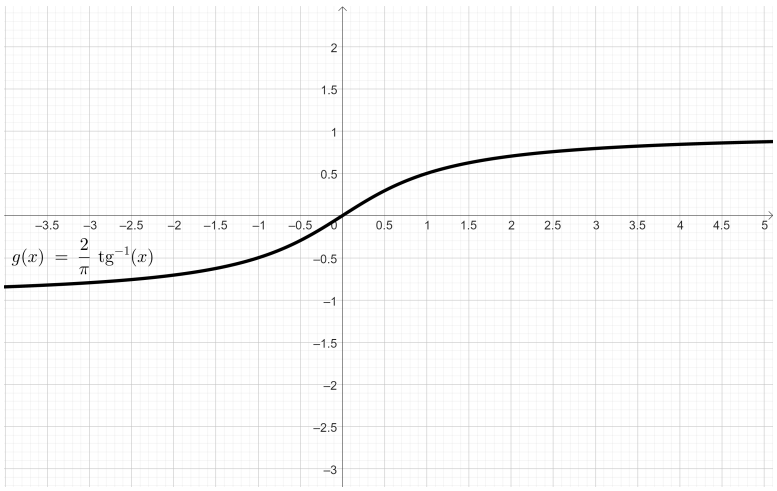

Figura 5.10: Gráfico de la función $\frac{2}{\pi} \arctan(x)$

Si bien las funciones de activación toman un papel bastante importante dentro de la teoría de redes neuronales, no son las únicas funciones en las que es necesario centrarse. También es importante recalcar que las Redes Neuronales buscan aproximar soluciones, o en algunos casos funciones, por lo que debemos tener un criterio para medir esta aproximación. En particular, se utiliza lo que se conoce como una función de pérdida o de costo, que nos permite cuantificar las aproximaciones y qué tan cercana a una solución se encuentra una red.

5.3. Ejemplos de funciones de pérdida

5.3.1. La función de error del supremo

Suponga que una Red Neuronal, que toma entradas $x \in I_n = [0,1]^n$ quiere aproximar una función continua $\phi : [0,1]^n \to \mathbb{R}$, y si $f_{w,b}$ es un mapeo de entrada y salida de la red entonces, la función de pérdida está dada por (Calin, 2020):

$$\mathcal{C}(w,b) = \sup_{x \in [0,1]^n} |f_{w,b}(x) - \phi(x)|$$

5.3.2. La función de error cuadrático medio

Considere una red que tiene como entrada una variable aleatoria X, y como salida una variable aleatoria $Y = f_{w,b}(X)$ donde $f_{w,b}$ es un mapeo de entrada y salida de la red que depende de los parámetros w y b para aproximar la variable aleatoria objetivo Z. Sea n el número mediciones de las variables aleatorias (X, Z) las cuales están dadas por pares ordenados $(x_1, z_1), \ldots (x_n, z_n)$, esto es lo que genera un conjunto de entrenamiento para una red, entonces, en este caso la función de pérdida está dada por (Calin, 2020):

$$\mathcal{C}(w,b) = \frac{1}{n} \sum_{i=1}^{m} \left(f_{w,b}(x_i) - z_i\right)^2$$

Nota: este tipo de función de pérdida es bastante útil en problemas de regresión.

5.3.3. Cross-entropy

Sea p y q dos densidades de $\mathbb{R}$ o cualquier otro intervalo. Se sabe que la función de verosimilitud negativa está dada por $-\mathcal{L}_q = -\ln(q(x))$, y esta mide la información brindada por $q(x)$. Entonces, la *cross - entropy* de p con respecto a q se define como (Calin, 2020):

$$S(p,q) = \mathbb{E}^p[-\mathcal{L}_q] = -\int_{\mathbb{R}} p(x) \ln(q(x)) dx$$

Nota: este tipo de función de pérdida es bastante útil en problemas de clasificación.

CAPÍTULO 6

La teoría de Cybenko

6.1. Conceptos preliminares

Para comenzar con esta versión del Teorema de Aproximación Universal, se define $I_n = [0,1]^n$ como el cubo $n-$ dimensional. El espacio de funciones continuas se denota de la siguiente manera $\mathcal{C}(I_n)$ y $||f||$ se refiere a la normal del supremo o norma uniforme de una función $f \in \mathcal{C}(I_n)$. De forma general, se utiliza $||\cdot||$ como referencia al máximo de una función particular sobre su dominio. Finalmente, tenemos que la nomenclatura de $M(I_n)$ alude al espacio de medidas regulares de Borel (12.4.5) sobre el hipercubo I_n (Cybenko, 1989), (Calin, 2020).

El objetivo del desarrollo de esta teoría es investigar las condiciones bajo las cuales una suma:

$$F(x) = \sum_{i=1}^{n} \alpha_i \cdot \sigma\left(w_i^{\intercal} \cdot x + b_i\right)$$

es densa sobre $\mathcal{C}(I_n)$ con respecto a la norma del supremo. Es necesario recordar que $w_i \in \mathbb{R}^n, \alpha_i, b_i \in \mathbb{R}$ y que $w^{\intercal} \cdot x$ representa el producto interno entre x y w. Además, se debe recalcar interés sobre el caso donde la función σ cumple con ser una función sigmoide:

$$\sigma(t) \to \begin{cases} 1, & t \to \infty \\ 0, & t \to -\infty \end{cases}$$

este tipo de funciones surge de forma natural durante el desarrollo de la teoría de Redes Neuronales como una función de activación sobre una neurona. (Cybenko, 1989)

Definición 6.1.1. ***Función sigmoide:*** *Se dice que una función diferenciable σ es sigmoide si cumple con lo siguiente:*

$$\sigma(t) \to \begin{cases} 1, & t \to \infty \\ 0, & t \to -\infty \end{cases}$$

a lo largo del trabajo para diferenciar una función arbitraria f de una función sigmoide se utilizará el símbolo σ para hacer referencia a esta última (Cybenko, 1989), (McNeela, 2017), (Calin, 2020).

Es necesario notar que la definición anterior no requiere que la función sea monótona. Sin embargo, si se establece que esta debe tener dos asíntotas horizontales. Un ejemplo de una función sigmoide es la función logística. También es importante recalcar que una medida μ puede ser considera como un sistema para guardar información(Calin, 2020). Por lo tanto, $d\mu(x) = \mu(dx)$ representa una evaluación de la información ingresada en x. De forma consiguiente, la integral:

$$\int f(x)d\mu$$

representa la evaluación de la función $f(x)$ bajo el sistema μ (Calin, 2020).

La siguiente definición introduce la notación de una función discriminatoria. Esta es una caracterización de la siguiente propiedad: si la evaluación de una neurona de salida ,$\sigma(w^\intercal x+b)$, sobre todas las posibles entradas de x, bajo la premisa que μ se desvanece para cualquier umbral b y cualesquiera pesos w, entonces μ se desvanece (Calin, 2020).

Definición 6.1.2. ***Función discriminatoria:*** *se dice que una función f es discriminatoria si para una medida $\mu \in M(I_n)$ se tiene que:*

$$\int_{I_n} f(w^\intercal \cdot x + b)d\mu(x) = 0$$

$\forall w \in \mathbb{R}^n$ y $\forall b \in \mathbb{R}$ implica que $\mu = 0$ (Cybenko, 1989), (Calin, 2020).

En esta definición es posible observar que f no es necesariamente una función sigmoide, sino cualquier función que cumpla con esta propiedad.

Definición 6.1.3. ***Partición de*** $\mathbb{R}^n$***:*** *Sea $\mathcal{P}_{w,b} = \{x|w^\intercal \cdot x + b = 0\}$ como el hiperplano con vector normal $w \in \mathbb{R}^n$ y $(n+1)-$ intercepto en $b \in \mathbb{R}$. De igual forma tenemos los semihiperspacios abiertos:*

$$\mathcal{H}^+_{w,b} = \{x|w^\intercal \cdot x + b > 0\}$$

$$\mathcal{H}^-_{w,b} = \{x|w^\intercal \cdot x + b < 0\}$$

los cuales son una partición del espacio $\mathbb{R}^n = \mathcal{H}^-_{w,b} \cup \mathcal{P}_{w,b} \cup \mathcal{H}^+_{w,b}$ (Calin, 2020).

Lema 6.1.1. *Sea $\mu \in M(I_n)$. Si μ se desvanece en todos los hiperplanos y semihiperspacios abiertos de $\mathbb{R}^n$ entonces $\mu = 0$. De forma precisa se tiene que:*

$$\mu(\mathcal{P}_{w,b}) = 0, \qquad \mu(\mathcal{H}^+_{w,b}) = \mu(\mathcal{H}^-_{w,b}) = 0, \qquad \forall w \in \mathbb{R}^n, \forall b \in \mathbb{R}$$

entonces $\mu = 0$ (Cybenko, 1989), (Calin, 2020).

Demostración. Sea $w \in \mathbb{R}^n$ un vector fijo y considere el funcional lineal $F : L^\infty(\mathbb{R}) \to \mathbb{R}$ definido por:

$$F(h) = \int_{I_n} h(w^\intercal \cdot x)d\mu(x)$$

donde $L^\infty(\mathbb{R})$ denota el espacio de funciones acotadas y medibles casi en todo punto sobre $\mathbb{R}$. Por lo siguiente podemos notar que F es acotado:

$$|F(h)| = \left|\int_{I_n} h(w^\intercal \cdot x)d\mu(x)\right| \leq ||h||_\infty \left|\int_{I_n} d\mu(x)\right| = ||h||_\infty \cdot |\mu(I_n)|$$

donde μ es finita porque I_n es un compacto (12.4.2).

Ahora si se toma $h = 1_{[b,\infty)}$ como la función indicador (12.1.1) sobre el intervalo $[b, \infty)$ y por la hipótesis de que μ se desvanece en todos los hiperplanos y semiplanos abiertos entonces:

$$F(h) = \int_{I_n} h(w^\intercal \cdot x) d\mu(x) = \int_{\{w^\intercal \cdot x \geq b\}} d\mu(x) = \mu(\mathcal{P}_{w,-b}) + \mu(\mathcal{H}^+_{w,-b}) = 0$$

Luego, si se toma $h = 1_{(b,\infty)}$ como la función indicador sobre el intervalo abierto (b, ∞), al computar se tiene:

$$F(h) = \int_{I_n} h(w^\intercal \cdot x) d\mu(x) = \int_{\{w^\intercal \cdot x > b\}} d\mu(x) = \mu(\mathcal{H}^+_{w,-b}) = 0$$

Ahora, por el hecho de que la función indicador de cualquier intervalo se puede reescribir como una combinación lineal de funciones indicador, es decir:

$$1_{[a,b]} = 1_{[a,\infty)} - 1_{(b,\infty)}, \qquad 1_{(a,b)} = 1_{(a,\infty)} - 1_{[b,\infty)}, \qquad a \leq b$$

Por la linealidad de F, se tiene que este se desvanece para cualquier función indicador. Al aplicar la linealidad nuevamente, obtenemos que esto en particular ocurre para funciones simples:

$$F\left(\sum_{i=1}^{N} \alpha_i 1_{J_i}\right) = \sum_{i=1}^{N} \alpha_i F(1_{J_i}) = 0$$

para cualquier $a_j \in \mathbb{R}$ y J_i un intervalo. Dado que la funciones simples son densas sobre $L^\infty(\mathbb{R})$ (12.4.8) tenemos que $F = 0$. De forma más específica, para cualquier función fija $f \in L^\infty(\mathbb{R})$, por la densidad de las funciones simples, debe existir una sucesión de funciones simples $(s_n)_n$ de tal forma que $(s_n) \to f$ cuando $n \to \infty$. Dado que F es acotado, entonces es continuo (12.5.2, (12.5.1) y por lo tanto, se tiene que:

$$F(f) = F(\lim_{n\to\infty} s_n) = \lim_{n\to\infty} F(s_n) = 0$$

En particular, se considera la Transformada de Fourier de la medida μ de la siguiente manera:

$$\begin{aligned}\hat{\mu}(w) = \int_{I_n} e^{iw^\intercal \cdot x} d\mu(x) &= \int_{I_n} \cos(w^\intercal \cdot x) d\mu(x) + i \int_{I_n} \sin(w^\intercal \cdot x) d\mu(x) \\ &= F(\cos(\cdot)) + iF(\sin(\cdot)) = 0\end{aligned}$$

dado que F se desvanece sobre las funciones acotadas seno y coseno. Por la inyectividad de la Transformada de Fourier se tiene que $\mu = 0$ (Cybenko, 1989), (Calin, 2020). □

Lema 6.1.2. *Cualquier función sigmoide es discriminatoria para todas las medidas $\mu \in M(I_n)$ (Cybenko, 1989), (Calin, 2020).*

Demostración. Sea $\mu \in M(I_n)$ una medida fija. Ahora sea σ una función sigmoide, la cual también es continua por la diferenciabilidad, que satisface:

$$\int_{I_n} \sigma(w^\intercal \cdot x + b) d\mu(x) = 0, \qquad \forall w \in \mathbb{R}^n, b \in \mathbb{R}$$

Es necesario probar que $\mu = 0$, para ello se construye la siguiente función continua con $x \in I_n, w \in \mathbb{R}^n, b, \phi \in \mathbb{R}, \lambda \in \mathbb{N}$ dados:

$$\sigma_\lambda(x) = \sigma(\lambda(w^\intercal \cdot x + b) + \phi)$$

Luego, por la definición de función sigmoide (6.1.1) se tiene que:

$$\lim_{\lambda\to\infty}\sigma_\lambda(x)=\begin{cases}1 & \text{para } w^\intercal x+b>0\\ 0 & \text{para } w^\intercal x+b<0\\ \sigma(\phi) & \text{para } w^\intercal x+b=0\end{cases}$$

Ahora se define la siguiente función acotada:

$$\gamma(x)=\begin{cases}1 & \text{si } x\in\mathcal{H}^+_{w,b}\\ 0 & \text{si } x\in\mathcal{H}^-_{w,b}\\ \sigma(\phi) & \text{si } x\in\mathcal{P}_{w,b}\end{cases}$$

y es importante ver que $\sigma_\lambda(x)\to\gamma(x)$ converge puntualmente sobre $\mathbb{R}$ cuando $\lambda\to\infty$ y que además, $|\sigma_\lambda(x)|\leq max(1,\sigma(\phi))$ para cualquier x. Notemos además, que $\{\sigma_\lambda(x)\}$ es una sucesión de funciones medibles por la continuidad de σ (12.4.5). Dado que σ es una función sigmoide sobre un compacto, esto implica que γ también es una función medible (12.4.6). Además, σ es integrable por ser continua sobre un compacto, y por consiguiente, es Lebesgue integrable (12.4.7). De esto se tiene que $\{\sigma_\lambda(x)\}$ es una sucesión de funciones integrables en el sentido de Lebesgue y además, es necesario notar que 1 y 0 son funciones continuas, al igual que $\sigma(\phi)$ por ser una función sigmoide, por lo que γ es una función medible e integrable, por lo que es integrable en el sentido de Lebesgue. De esta forma, es posible hacer uso del Teorema de Convergencia Dominada de Lebesgue (Cybenko, 1989), (Calin, 2020):

$$\begin{aligned}\lim_{\lambda\to\infty}\int_{I_n}\sigma_\lambda(x)d\mu(x)&=\int_{I_n}\lim_{\lambda\to\infty}\sigma_\lambda(x)d\mu(x)=\int_{I_n}\gamma(x)d\mu(x)\\ &=\int_{\mathcal{H}^-_{w,b}}\gamma(x)d\mu(x)+\int_{\mathcal{P}_{w,b}}\gamma(x)d\mu(x)+\int_{\mathcal{H}^+_{w,b}}\gamma(x)d\mu(x)\\ &=\sigma(\phi)\mu(\mathcal{P}_{w,b})+\mu(\mathcal{H}^+_{w,b})\end{aligned}$$

Ahora, por el hecho de que σ se seleccionó de tal forma que cumplan con:

$$\int_{I_n}\sigma(w^\intercal\cdot x+b)d\mu(x)=0,\qquad\forall w\in\mathbb{R}^n,b\in\mathbb{R}$$

entonces se tiene que:

$$\sigma(\phi)\mu(\mathcal{P}_{w,b})+\mu(\mathcal{H}^+_{w,b})=0$$

Dado que esto se cumple para cualquier valor de ϕ, en particular, es posible que $\phi\to\infty$ y por definición de una función sigmoide se tiene:

$$\mu(\mathcal{P}_{w,b})+\mu(\mathcal{H}^+_{w,b})=0,\qquad\forall w\in\mathbb{R}^n,b\in\mathbb{R}$$

y al tomar $\phi\to-\infty$ se obtiene que:

$$\mu(\mathcal{H}^+_{w,b})=0,\qquad\forall w\in\mathbb{R}^n,b\in\mathbb{R}$$

y por consiguiente

$$\mu(\mathcal{P}_{w,b})=0,\qquad\forall w\in\mathbb{R}^n,b\in\mathbb{R}$$

además, al establecer que:

$$\mu(\mathcal{H}^+_{w,b})=\mu(\mathcal{H}^-_{-w,-b})=0,\qquad\forall w\in\mathbb{R}^n,b\in\mathbb{R}$$

por el lema 6.1.1 se sabe que una medida μ se desvanece en un espacio con estas características, por lo que $\mu=0$ y por lo tanto σ es discriminatoria. (Cybenko, 1989), (Calin, 2020), (McNeela, 2017).

□

6.2. Aprendiendo funciones continuas $f \in \mathcal{C}(I_n)$

Dentro de esta sección se demuestra que una Red Neuronal de una sola capa oculta puede ser utilizada para aproximar funciones sobre le espacio $\mathcal{C}(I_n)$. Esto quiere decir que dada una función $f(x)$ que pertenece a este espacio, y con la variable de entrada $x \in I_n$ entonces hay una combinación de los pesos de una red de tal forma que el resultado $G(x)$ es arbitrariamente cercano al de la función $f(x)$, bajo una cierta distancia. En otras palabras, dada una suma finita de la forma:

$$G(x) = \sum_{i=1}^{N} \alpha_i \sigma(w_i^\intercal \cdot x + b_i)$$

para ciertos valores α_i, w_i, b_i, dado $\epsilon > 0$ y $g \in \mathcal{C}(I_m)$ podemos tener la siguiente desigualdad:

$$|f(x) - G(x)| < \epsilon, \qquad \forall x \in I_n$$

Antes de avanzar a los resultados principales, es necesario contar con resultados preliminares de Análisis Funcional. El primer lema garantiza la existencia de un funcional de separación, el cual se desvanece sobre un subespacio y tiene un valor distinto de cero fuera del subespacio (Cybenko, 1989), (Calin, 2020).

Lema 6.2.1. *Sea $\mathcal{U}$ un subespacio lineal de un espacio lineal normado X y considere $x_0 \in X$ de tal forma que:*

$$dist(x_0, \mathcal{U}) \geq \delta$$

para algún $\delta > 0$, es decir, $|||x_0 - u|| \geq \delta$ para todo $u \in \mathcal{U}$. Entonces, existe un funcional lineal L sobre X de tal forma que (Calin, 2020):

1. $||L|| \leq 1$
2. $L(u) = 0, \qquad \forall u \in \mathcal{U}$
3. $L(x_0) = \delta$

Demostración. Considere el espacio lineal T generado por $\mathcal{U}$ y x_0:

$$T = \{t | t = u + \lambda x_0, \quad u \in \mathcal{U}, \quad \lambda \in \mathbb{R}\}$$

y se define la función $L : T \to \mathbb{R}$ de la siguiente manera:

$$L(t) = L(u + \lambda x_0) = \lambda \delta$$

Ahora, sea $t_1, t_2 \in T$ y $\alpha \in \mathbb{R}$ entonces tenemos que:

$$L(t_1 + t_2) = L(u_1 + u_2 + (\lambda_1 + \lambda_2)x_0) = (\lambda_1 + \lambda_2)\delta = \lambda_1\delta + \lambda_2\delta$$
$$= L(u_1 + \lambda_1 x_0) + L(u_2 + \lambda_2 x_0) = L(t_1) + L(t_2)$$

y también:

$$L(\alpha t) = L(\alpha(u + \lambda x_0)) = L(\alpha u + \alpha\lambda x_0) = \alpha\lambda\delta = \alpha L(t)$$

por lo que L es un funcional lineal sobre T.

Ahora, se demostrará que $||L|| \leq ||t||$ para todo $t \in T$. Dado que U es un subespacio lineal, por lo que $u \in U$ implica que $-\frac{u}{\lambda} \in U$. Por tanto:

$$||x_0 + \frac{u}{\lambda}|| = ||x_0 - \left(-\frac{u}{\lambda}\right)|| \geq \delta$$

o de forma equivalente tenemos la siguiente desigualdad:

$$||x_0 + \frac{u}{\lambda}|| \geq \delta \Longrightarrow \frac{1}{||x_0 + \frac{u}{\lambda}||} \leq \frac{1}{\delta} \Longrightarrow \frac{\delta}{||x_0 + \frac{u}{\lambda}||} \leq 1 \Longrightarrow |\lambda|\delta \leq ||u + \lambda x_0||$$

por lo tanto:

$$L(t) = L(u + \lambda x_0) = \lambda\delta \leq |\lambda|\delta \leq ||u + \lambda x_0|| = ||t||$$

para todo $t \in T$. Al aplicar el teorema de Hahn - Banach (versión 2 12.5.3) con la norma $p(x) = ||x||$ tenemos que L puede extender a un funcional lineal sobre X, denotado también por L, de tal forma que $L(x) \leq ||x||$ para todo $x \in X$. Esto implica que $||L|| \leq 1$, por lo que la extensión de L es acotada. Además, por definición tenemos lo siguiente:

$$L(u) = L(u + 0x_0) = 0 \cdot \delta, \qquad \forall u \in \mathcal{U}$$
$$L(x_0) = L(0 + 1 \cdot x_0) = 1 \cdot \delta, \qquad \delta > 0$$

□

El resultado anterior puede ser reformulado en términos de un subespacio denso. Un subespacio $\mathcal{U}$ es denso en X con respecto a la norma $||\cdot||$ si para cualquier elemento $x \in X$ hay elementos $u \in \mathcal{U}$ bastante cercanos a x. De manera equivalente, para todo $x \in X$ existe una sucesión $(u_n) \in \mathcal{U}$ de tal forma que $(u_n) \to x$ cuando $n \to \infty$, o bien, para todo $x \in X$, $\epsilon > 0$ existe $u \in \mathcal{U}$ de tal forma que $||u - x|| < \epsilon$. Consecuentemente, el hecho de que un subespacio $\mathcal{U}$ no sea denso en X puede describirse como: existen elementos $x_0 \in X$ de tal forma que no hay elementos $u \in \mathcal{U}$ no sean lo suficientemente cercanos a x_0. Es decir, existe $\delta > 0$ de tal forma que para todo $u \in \mathcal{U}$ tenemos $||u - x_0|| \geq \delta$, esta es la hipótesis del lema 6.2.1, por lo que tenemos el siguiente lema:

Lema 6.2.2. ***Lema de Reformulación****: sea $\mathcal{U}$ un subespacio lineal no denso del espacio lineal normado X. entonces, existe un funcional lineal acotado L en X de tal forma que $L \neq 0$ sobre $X \setminus \mathcal{U}$ y $L(\mathcal{U}) = 0$ (Calin, 2020).*

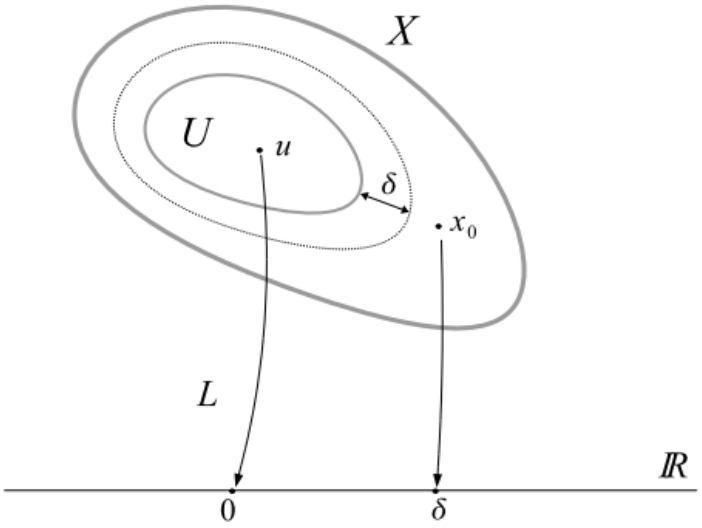

Figura 6.1: Funcional Lineal del lema 6.2.2.

Ahora, denote $\mathcal{C}(I_n)$ el espacio lineal de funciones continuas sobre el hipercubo $I_n = [0, 1]^n$ como un espacio normado con la norma:

$$||f|| = \sup_{x \in I_n} |f((x)|, \qquad \forall f \in \mathcal{C}(I_n)$$

Sea $M(I_n)$ el espacio de medidas finitas de Borel sobre I_n. Es necesario notar que también puede trabajarse con medidas de Baire, dado que se está trabajando sobre un espacio métrico compacto, esto implica que los conjuntos de Baire y los conjuntos de Borel son los mismos, y por tanto, las medidas finitas coinciden en los conjuntos compactos.
El siguiente resultado muestra que siempre es posible encontrar una medida que se desvanece sobre un subespacio no denso de $\mathcal{C}(I_n)$ (Calin, 2020).

Lema 6.2.3. *Sea $\mathcal{U}$ un subespacio lineal no denso de $\mathcal{C}(I_n)$ entonces, existe una medida $\mu \in M(I_n)$ de tal forma que:*

$$\int_{I_n} h d\mu = 0, \qquad \forall h \in \mathcal{U}$$

Demostración. Considere $X = \mathcal{C}(I_n)$, por el lema 6.2.2 existe un funcional lineal $L : \mathcal{C}(I_n) \to \mathbb{R}$ de tal forma que $L \neq 0$ sobre $\mathcal{C}(I_n) \setminus \mathcal{U}$ y además $L(\mathcal{U}) = 0$. Por el teorema de Representación de Riesz (versión 2 12.4.9) aplicado a L sabemos que existe una medida $\mu \in M(I_n)$ que cumple con:

$$L(f) = \int_{I_n} f d\mu, \qquad \forall f \in \mathcal{C}(I_n).$$

En particular, para cualquier $h \in \mathcal{U}$ obtenemos que (Calin, 2020):

$$L(h) = \int_{I_n} h d\mu = 0$$

NOTA: $L \neq 0$ sobre $\mathcal{C}(I_n) \setminus \mathcal{U} \Rightarrow \mu \neq 0$ sobre $\mathcal{C}(I_n) \setminus \mathcal{U}$ □

6.3. Teorema de aproximación universal (versión 1)

Teorema 6.3.1. *Sea σ una función continua y discriminatoria en el sentido de la definición 6.1.2. Entonces, las sumas finitas de la forma:*

$$G(x) = \sum_{i=1}^{N} \alpha_i \sigma(w_i^\intercal \cdot x + b_i), \qquad \forall w_i \in \mathbb{R}^n, \alpha_i, b_i \in \mathbb{R}$$

son densas en $\mathcal{C}(I_n)$

Demostración. Dado que σ es una función continua entonces sabemos que:

$$\mathcal{U} = \left\{ G \middle| G(x) = \sum_{i=1}^{N} \alpha_i \sigma(w_i^\intercal \cdot x + b_i) \right\}$$

es un subespacio lineal de $\mathcal{C}(I_n)$. La prueba se completa por contradicción. Supóngase que $\mathcal{U}$ no es denso en $\mathcal{C}(I_n)$, por el lema 6.2.3 se sabe que existe una medida $\mu \in M(I_n)$ de tal forma que

$$\int_{I_n} h d\mu = 0, \qquad \forall h \in \mathcal{U}$$

lo cual se puede reescribir como

$$\sum_{i=1}^{N} \alpha_i \int_{I_n} \sigma(w_i^\intercal \cdot x + b_i) d\mu = 0, \qquad \forall w_i \in \mathbb{R}^n, b_i, \alpha_i \in \mathbb{R}$$

al escoger los coeficientes adecuados de α_i se tiene:

$$\int_{I_n} \sigma(w^\intercal \cdot x + b) d\mu = 0, \qquad \forall w \in \mathbb{R}^n, b \in \mathbb{R}$$

y se sabe que σ es discriminatoria, por lo que esto implica que $\mu = 0$, lo cual es una contradicción dado que por el lema 6.2.3 $L \neq 0$ sobre $\mathcal{C}(I_n) \setminus \mathcal{U} \Rightarrow \mu \neq 0$ sobre $\mathcal{C}(I_n) \setminus \mathcal{U}$. Por tanto, $\mathcal{U}$ es denso en $\mathcal{C}(I_n)$(*Cybenko*,1989), (*Calin*,2020). □

El resultado anterior nos permite establecer que para toda función $f \in \mathcal{C}(I_n)$ y dado $\epsilon > 0$ existe una suma finita $G(x)$ de tal forma que:

$$|G(x) - f(x)| < \epsilon, \qquad \forall x \in I_n$$

De esto se tiene que la Red Neuronal de una sola capa puede aprender cualquier función continua sobre I_n, con un error pequeño usando los pesos adecuados. El siguiente resultado indica que, al no restringir la cantidad de nodos y el tamaño de los pesos de una Red Neuronal de una sola capa oculta esta es un aproximador universal. Las entradas son los vectores $x^\intercal = (x_1, \ldots, x_n) \in I_n$ y los pesos de las entradas de la capa oculta se denotan $(w_1, \ldots, w_n)$ donde cada peso es un vector de la forma $w_i^\intercal = (w_{i1}, \ldots, w_{in})$. Los pesos de la capa oculta a la función de salida son $(\alpha_1, \ldots, \alpha_N)$ con $\alpha_i \in \mathbb{R}$. El número de neuronas dentro de la capa oculta es N, y los sesgos están dados por $(b_1, \ldots, b_N)$. Debemos notar que la función de activación de la capa de salida es lineal $\phi(x) = x$, mientras que en la capa oculta es sigmoide (Cybenko, 1989), (Calin, 2020).

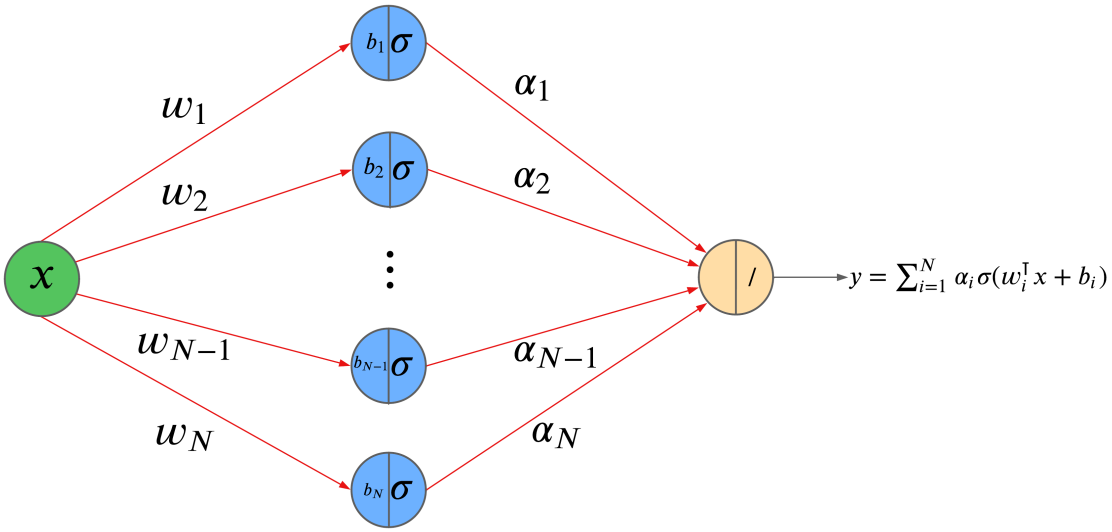

Figura 6.2: Diagrama de una red neuronal de una sola capa oculta de N nodos con una función de activación sigmoide.

Teorema 6.3.2. ***Teorema de Aproximación Universal****: sea σ una función sigmoide en el sentido de la definición 6.1.1. Entonces, las sumas finita de la forma:*

$$G(x) = \sum_{i=1}^{N} \alpha_i \sigma(w_i^\intercal \cdot x + b_i), \qquad \forall w_i \in \mathbb{R}^n, \alpha_i, b_i \in \mathbb{R}$$

son densas en $\mathcal{C}(I_n)$ (Cybenko, 1989), (Calin, 2020).

Demostración. Por el lema 6.1.2 sabemos que la función σ es discriminatoria, y al aplicar el resultado del teorema 6.3.1 se tiene el resultado deseado. □

6.4. ¿Son suficientes las funciones sigmoide?

Es importante hacer énfasis en lo siguiente, sea σ la función sigmoide definida por:

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

entonces todas las funciones de $G(x)$ son analíticas, por ser una suma finita de funciones analíticas. Sin embargo, sabemos que $\mathcal{C}(I_n)$ es un espacio de funciones continuas, por lo tanto hay funciones en ese espacio que no son analíticas. De esto se tiene que el espacio de aproximación $\mathcal{U}$ es un subespacio propio de $\mathcal{C}(I_n)$, lo que nos indica que las funciones sigmoide no son las únicas funciones de activación que se pueden utilizar por lo que vimos en la sección 5.2. En el siguiente capítulo se trabaja una versión más general del Teorema de aproximación Universal (Cybenko, 1989), (Calin, 2020).

CAPÍTULO 7

La teoría de Hornik, Stinchcombe y White

En este capítulo se trabaja una versión más general para el Teorema de Aproximación Universal, en este caso, solo se necesita de una función de activación continua y no constante. Sin embargo, hay un precio que se debe pagar para poder llegar a esta generalización. En el capítulo 6 se trabaja con sumas finitas de funciones, esto se puede denominar como redes $\Sigma-$ clase, sin embargo, en el caso más general es necesario sustituir esta clase particular de funciones por una denominada redes $\Pi\Sigma-$ clase, o bien el producto finito de las sumas finitas.

7.1. Conceptos preliminares

Definición 7.1.1. ***Retículo****: sea $L \subseteq \mathcal{C}(X)$, donde $\mathcal{C}(X)$ denota el espacio de funciones continuas sobre un espacio compacto de Hausdorff X. Considere las siguientes definiciones de máximo y mínimo de funciones:*

$$f \wedge g = (f \wedge g)(x) = \text{mín}(f(x), g(x))$$
$$f \vee g = (f \vee g)(x) = \text{máx}(f(x), g(x))$$

A L se le llama retículo si para todo par de funciones $f, g \in L$ implica que $f \wedge g, f \vee g \in L$. En otras palabras, tenemos que para cada par de elementos en L podemos encontrar una cota superior y una cota inferior (Royden, 1988).

Lema 7.1.1. *Sea L un retículo, dado por la definición 7.1.1, de funciones continuas sobre un espacio compacto de Hausdorff X y supóngase que la función definida por:*

$$h(x) = \inf_{f \in L} f(x)$$

es continua. Entonces, dado $\epsilon > 0$ existe una función $g \in L$ de tal forma que $0 \leq g(x) - h(x) < \epsilon$ para todo $x \in X$ (Royden, 1988).

Demostración. Dado $\epsilon > 0$ y, para todo $x \in X$ existe una función $f_x \in L$ de tal forma que $f_x(x) < h(x) + \epsilon/3$ debido a que $h(x)$ se define como el ínfimo del retículo de funciones continuas sobre un

espacio compacto de Hausdorff evaluadas en un punto x, por lo que esto garantiza la existencia del $\epsilon > 0$ que genera la desigualdad anterior. Además, se sabe que f_x y $h(x)$ son funciones continuas, entonces existe un conjunto abierto B_x que contiene a x de tal forma que:

$$|f_x(y) - f_x(x)| < \epsilon/3 \qquad \text{y} \qquad |h(y) - h(x)| < \epsilon/3$$

para todo $y \in B_x$. De esto tenemos lo siguiente:

$$f_x(y) - h(y) \leq |f_x(y) - h(y)| = |f_x(y) - f_x(x) + f_x(x) - h(x) + h(x) - h(y)|$$

$$\leq |f_x(y) - f_x(x)| + |f_x(x) - h(x)| + |h(x) - h(y)| < \epsilon/3 + \epsilon/3 + \epsilon/3 = \epsilon$$

y por ende:

$$f_x(y) - h(y) < \epsilon$$

para todo $y \in B_x$. Ahora, los conjuntos B_x por ser abiertos, forman una subcubierta finita sobre X por definición de compacidad, $\{B_{x_1}, \ldots, B_{x_n}\}$. Se toma $g = f_{x_1} \wedge f_{x_2} \wedge \cdots \wedge f_{x_n}$, y por que L es un retículo, sabemos que $g \in L$, y dado $y \in X$ se escoge i de tal forma que $y \in B_{x_i}$, y por lo tanto:

$$0 \leq g(y) - h(y) \leq f_{x_i}(y) - h(y) < \epsilon$$

□

Lema 7.1.2. *Sea X un espacio compacto de Hausdorff y sea L un retículo de funciones continuas sobre X con las siguientes propiedades (Royden, 1988):*

1. *L separa puntos, es decir, si $x \neq y$ entonces existe $f \in L$ de tal forma que $f(x) \neq f(y)$*
2. *si $f \in L$, y c es un número real, entonces cf y $c + f$ también pertenecen a L*

Entonces dada cualquier función continua h sobre X y para todo $\epsilon > 0$ existe una función $g \in L$ de tal forma que para todo $x \in X$ se tiene que:

$$0 \leq g(x) - h(x) < \epsilon$$

Antes de empezar con la demostración se probaron dos lemas:

Lema 7.1.3. ***Auxiliar 1:*** *Sea L una familia de funciones sobre un espacio compacto X que satisfacen las condiciones 1. y 2. del lema 7.1.2. Entonces para cualesquiera números reales $a, b \in \mathbb{R}$ y cualquier par x, y de puntos distintos en X existe una función $f \in L$ de tal forma $f(x) = a$ y $f(y) = b$ (Royden, 1988).*

Demostración. Sean $a, b \in \mathbb{R}$, y considere x, y dos puntos distintos en X, es decir $x \neq y$, entonces por la condición 1. sabemos que existe $g \in L$ de tal forma que $g(x) \neq g(y)$, lo cual permite construir una función $f \in L$ de la siguiente forma:

$$f = \frac{a - b}{g(x) - g(y)} \cdot g + \frac{bg(x) - ag(y)}{g(x) - g(y)},$$

por la condición 2. Y por lo siguiente $f(x) = a$ y $f(y) = b$:

$$f(x) = \frac{ag(x) - bg(x) + bg(x) - ag(y)}{g(x) - g(y)} = a$$

$$f(y) = \frac{ag(y) - bg(y) + bg(x) - ag(y)}{g(x) - g(y)} = b$$

□

Lema 7.1.4. ***Auxiliar 2:*** *Sea L un retículo que cumple con las condiciones del lema 7.1.2, a y b números reales con $a \leq b$, F un subconjunto cerrado de X, y p un punto que no esté en F. Entonces, existe una función $f \in L$ de tal forma que $f \geq a, f(p) = a$ y $f(x) > b$ para todo $x \in F$ (Royden, 1988).*

Demostración. Sea L un retículo que cumple con las condiciones del lema 7.1.2, a y b números reales con $a \leq b$, F un subconjunto cerrado de X, y p un punto que no esté en F. Por el lema 7.1.3 podemos escoger, para cada $x \in F$ una función f_x de tal forma que $f_x(p) = a$ y $f_x(x) = b + 1$. Sea $B_x = \{y : f_x(y) > b\}$, entonces, los conjuntos $\{B_{x_1}, \ldots, B_{x_n}\}$ son una cubierta finita (12.3.1) de F porque F es compacto en su topología relativa (12.3.2). Considere, $f = f_{x_1} \vee \cdots \vee f_{x_n}$, entonces $f \in L$, $f(p) = a$ y $f > b$ para todo $x \in F$, al reemplazar f con $f \vee a$ se cumple $f \geq a$ sobre X □

Demostración. Lema 7.1.2:
Por la condición 1. se sabe que L es no vacío debido a que contiene al menos una función f que separa puntos. Luego, por la condición 2. se puede escoger $c = 0$ y dado $f \in L$ se tiene $0 \cdot f = 0 \in L$, por lo que la función constante $h(x) = 0, \forall x \in X$ con $h \in L$. Con esto, es posible generar el resto de funciones constante al aplicar la condición 2. sobre h para una constante real c arbitraria. Ahora, dada una función $g \in \mathcal{C}(X)$ y sea $L' = \{f : f \in L, f \geq g\}$. Esta demostración es directa del lema 7.1.1 si se demuestra que para todo $p \in X$ se tiene que $g(p) = \text{ínf}\, f(p)$ donde $f \in L'$.
Considere un número real positivo a, dado que g es continua entonces el conjunto:

$$F = \{x : g(x) \geq g(p) + a\}$$

es cerrado, y dado que X es un espacio compacto de Hausdorff entonces g es acotada sobre X por una constante M, es necesario notar que siempre se puede encontrar M tal que $g(p) + a \leq M$ porque a es arbitrario. Por el lema 7.1.4 es posible encontrar una función $f \in L$ de tal forma que $f \geq g(p) + a$, $f(p) = g(p) + a$ y $f(x) > M$ para todo $x \in F$. Luego, se sabe que $g < g(p) + a$ sobre F^{c} lo que implica que $g < f$ sobre X, entonces $f \in L'$, y por ende, $f(p) \leq g(p) + a$, dado que a es un número positivo entonces podemos garantizar la existencia de un f que cumpla la desigualdad para cada a arbitrariamente pequeño, por lo que $g(p) = \text{ínf}\, f(p)$ para $f \in L'$.

□

Lema 7.1.5. *Dado $\epsilon > 0$ existe un polinomio P en una variable de tal forma que para todo $s \in [-1, 1]$ tenemos que $|P(s) - |s|| < \epsilon$ (Royden, 1988).*

Demostración. Sea $\sum_{n=0}^{\infty} c_n t^n$ la serie binomial para $(1-t)^{1/2}$. Esta serie converge de forma uniforme para valores de $t \in [0, 1]$ (12.2.2, 12.2.1). Por lo tanto, dado $\epsilon > 0$ se puede escoger N de tal forma que $t \in [0, 1]$ se tiene:

$$|(1 - t)^{1/2} - Q_N(t)| < \epsilon$$

donde

$$Q_N(t) = \sum_{n=0}^{N} c_n t^n.$$

Sea $P(s) = Q_N((1 - s)(1 + s))$ entonces P es un polinomio en función de s y se tiene que

$$||s| - P(s)| \leq |\sqrt{s^2} - Q_N(1 - s^2)| = |\sqrt{1 - t} - Q_N(t)| < \epsilon$$

para $t = 1 - s^2, s \in [-1, 1]$. □

7.2. Teorema de Stone - Weierstrass

Teorema 7.2.1. ***Teorema de Stone - Weierstrass:***

Sea X un espacio compacto de Hausdorff y A un álgebra de funciones reales y continuas sobre X que separan puntos de X y contienen funciones constantes. Entonces, dada cualquier función continua f sobre X y dado $\epsilon > 0$ existe una función $g \in A$ de tal forma que para todo $x \in X$ se tiene que

$$|g(x) - f(x)| < \epsilon$$

en otras palabras, A es un subconjunto denso de $\mathcal{C}(X)$ (Royden, 1988).

Demostración. Sea $\overline{A} \subset \mathcal{C}(X)$ la cerradura de A. Entonces $\overline{A}$ consiste en las funciones sobre X que son los límites de una sucesión de funciones generada en A. Esta convergencia es uniforme porque sabemos que A contiene funciones continuas sobre un compacto. Dado que $\overline{A} \subset \mathcal{C}(X)$ entonces tenemos que para $f, g \in \overline{A}$ entonces existen $f_n, g_n \in A$ de tal forma que $f_n \to f$ y $g_n \to g$, uniformemente. Entonces, por la convergencia uniforme se tiene que $f_n + g_n \to f + g$, $f_n g_n \to fg$, $\lambda f_n \to \lambda f$ para $\lambda \in \mathbb{R}$, por lo que entonces se tiene que

$$f + g \in \overline{A}, \qquad fg \in \overline{A}, \qquad \lambda f \in \overline{A} \quad \lambda \in \mathbb{R}$$

por lo que $\overline{A}$ es un álgebra (Bernués, 2010). Lo que se quiere demostrar es que $\overline{A} = \mathcal{C}(X)$, esto ocurre directamente del lema 7.1.2 si se prueba que $\overline{A}$ es un retículo. Entonces, sea $f \in A$ y $||f|| \leq 1$ lo que implica que $\sup |f(x)| \leq 1$ para todo $x \in X$. Por ende, $|f(x)| \leq 1$ para todo $x \in X$. Dado $\epsilon > 0$ se tiene que:

$$|||f| - P(f)|| = \sup_{|f(x)| \leq 1, \quad x \in X} ||f(x)| - P(f(x))| \leq \sup_{|s| \leq 1} ||s| - P(s)| < \epsilon$$

donde P es el polinomio dado por el Lema 7.1.5. Dado que $\overline{A}$ es un álgebra que contiene las funciones constantes entonces $P(f) \in A$, y dado que $\overline{A}$ es un subconjunto cerrado de $\mathcal{C}(X)$ entonces tenemos que $|f| \in \overline{A}$. Ahora, si $f \in A$ es una función cualquiera, entonces $f/||f||$ tiene norma igual a 1, y por ende $|f|/||f||$ tiene una norma igual a 1, por lo tanto, $|f| \in \overline{A}$. Entonces $\overline{A}$ contiene el valor absoluto de cada una de las funciones en $\overline{A}$, entonces:

$$f \vee g = \frac{1}{2}(f + g) + \frac{1}{2}|f - g|$$

y

$$f \wedge g = \frac{1}{2}(f + g) - \frac{1}{2}|f - g|.$$

Por lo que $\overline{A}$ es un retículo y debe cumplirse que $\overline{A} = \mathcal{C}(X)$ por el lema 7.1.2. □

7.3. Aplicación del Teorema de Stone - Weierstrass en las Redes Neuronales

Una de las aplicaciones de este teorema es que las Redes Neuronales con funciones de activación de seno y coseno pueden aprender cualquier función periódica (Calin, 2020).
Considere el siguiente ejemplo de una Red Neuronal de una sola capa oculta con entradas reales x, una salida unidimensional y y N neuronas en la capa oculta. Se toma la función de activación como $\phi(x) = \cos(x)$, esta función es continua y analítica en $\mathbb{R}$, pero no es una función sigmoide, por lo que ninguno de los resultados vistos en la sección 6 es aplicable. En este caso la red está dada por:

$$y = a_0 + \sum_{i=1}^{N} \alpha_i \cos(w_i x + b_i)$$

donde w_i, α_i son los pesos de la entrada a la capa oculta y de la capa oculta a la salida, respectivamente. Los sesgos en la capa oculta se denotan como b_i mientras que a_0 hace referencia al sesgo de la neurona de salida. Además, considere la siguiente función continua $f : \mathbb{R} \to \mathbb{R}$ la cual es periódica con periodo T, lo que significa que $f(x+T) = f(x)$. Sea $v = \frac{2\pi}{T}$ la frecuencia asociada y considere los pesos $w_i = iv$ de lo que se tiene :

$$\begin{aligned} y &= a_0 + \sum_{i=1}^{N} \alpha_i \cos(ivx + \beta_i) \\ &= a_0 + \sum_{i=1}^{N} \alpha_i \cos(ivx)\cos(\beta_i) - \alpha_i \sin(ivx)\sin(\beta_i) \\ &= a_0 + \sum_{i=1}^{N} a_i \cos(ivx) + b_i \sin(ivx) \\ &= a_0 + \sum_{i=1}^{N} a_i \cos\left(\frac{2\pi ix}{T}\right) + b_i \sin\left(\frac{2\pi ix}{T}\right) \end{aligned}$$

donde $a_i = \alpha_i \cos(\beta_i)$ y $b_i = -\alpha_i \sin(\beta_i)$. Esto debe ser familiar, ya que es una suma finita de coeficientes de Fourier. Ahora, sabemos que la Serie de Fourier es un buen aproximador para funciones periódicas, en particular para funciones periódicas y continuas. Por lo tanto, hemos encontrado un caso específico de una función de activación analítica no sigmoide que puede aproximar bastante bien funciones continuas. Por ello, el resultado del teorema 6.3.2 no es lo suficientemente general. En la siguiente sección se trabajará con funciones de activación más generales, en particular, funciones de activación continuas no constantes. (Calin, 2020).

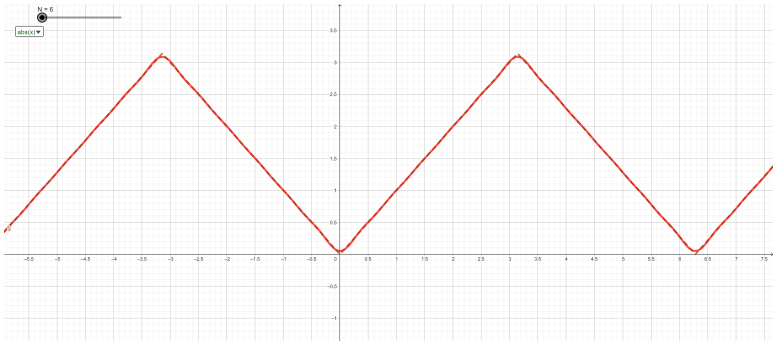

Figura 7.1: Serie finita de Fourier (N = 6) aproximando una función valor absoluto periódica continua.

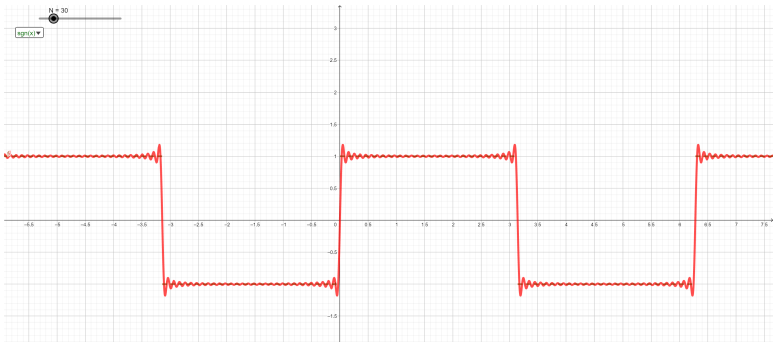

Figura 7.2: Serie finita de Fourier (N = 30) aproximando una función signo periódica.

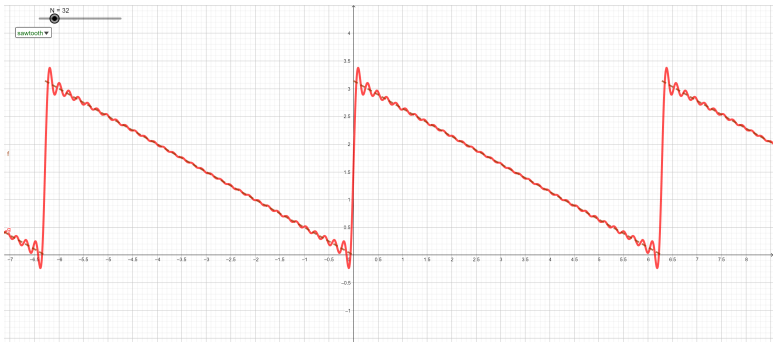

Figura 7.3: Serie finita de Fourier (N = 32) aproximando una función diente de cierra periódica.

7.4. Teorema de aproximación universal (versión 2)

Teorema 7.4.1. ***Teorema de Aproximación Universal:***

Sea $\varphi : \mathbb{R} \to \mathbb{R}$ cualquier función continua de activación no constante. Entonces, las sumas finitas de los productos finitos de la forma:

$$G(x) = \sum_{i=1}^{M} \beta_i \prod_{j=1}^{N_i} \varphi(w_{ij}^{\intercal} x + b_{ij})$$

donde $w_{ij} \in \mathbb{R}^n$, $\beta_i, b_{ij} \in \mathbb{R}, x \in I_n, M, N_i = 1, 2, \ldots$ son densas en $\mathcal{C}(I_n)$ (Calin, 2020), (Hornik, Stinchcombe, y White, 1989).

Demostración. Considere el conjunto $\mathcal{U}$ dado por las sumas finitas de los productos finitos de la forma:

$$G(x) = \sum_{i=1}^{M} \beta_i \prod_{j=1}^{N_i} \varphi(w_{ij}^{\intercal} x + b_{ij})$$

donde $w_{ij} \in \mathbb{R}^n$, $\beta_i, b_{ij} \in \mathbb{R}, x \in I_n, M, N_i = 1, 2, \ldots$

Dado que φ es una función continua no constante, entonces se tiene que para cualesquiera $G, G_1, G_2 \in \mathcal{U}$ dados por la forma:

$$G(x) = \sum_{i=1}^{M} \beta_i \prod_{j=1}^{N_i} \varphi(w_{ij}^{\intercal} x + b_{ij})$$

$$G_1(x) = \sum_{i=1}^{K} \alpha_i \prod_{j=1}^{L_i} \varphi(v_{ij}^{\intercal} x + a_{ij})$$

$$G_2(x) = \sum_{i=1}^{T} \gamma_i \prod_{j=1}^{P_i} \varphi(u_{ij}^{\intercal} x + c_{ij})$$

se satisface lo siguiente:

- $\lambda G \in \mathcal{U}, \qquad \forall \lambda \in \mathbb{R}$:

Sea $\lambda \in R$ por lo que:

$$\lambda G(x) = \lambda \sum_{i=1}^{M} \beta_i \prod_{j=1}^{N_i} \varphi(w_{ij}^{\intercal} x + b_{ij}) = \sum_{i=1}^{M} \lambda\beta_i \prod_{j=1}^{N_i} \varphi(w_{ij}^{\intercal} x + b_{ij}) = \sum_{i=1}^{M} \beta_i' \prod_{j=1}^{N_i} \varphi(w_{ij}^{\intercal} x + b_{ij})$$

- $G_1 + G_2 \in \mathcal{U}$:

$$G_1(x) + G_2(x) = \sum_{i=1}^{K} \alpha_i \prod_{j=1}^{L_i} \varphi(v_{ij}^{\intercal} x + a_{ij}) + \sum_{i=1}^{T} \gamma_i \prod_{j=1}^{P_i} \varphi(u_{ij}^{\intercal} x + c_{ij}) = \sum_{i=1}^{K+T} \beta_i'$$

donde:

$$\beta_i' = \begin{cases} \alpha_i \prod_{j=1}^{L_i} \varphi(v_{ij}^{\intercal} x + a_{ij}), & 1 \leq i \leq K \\ \gamma_{i-K} \prod_{j=1}^{P_{i-K}} \varphi(u_{(i-K)j}^{\intercal} x + c_{(i-K)j}), & K+1 \leq i \leq K+T \end{cases} \tag{7.1}$$

- $G_1 G_2 \in \mathcal{U}$:

$$G_1(x)G_2(x) = \left(\sum_{i=1}^{K} \alpha_i \prod_{j=1}^{L_i} \varphi(v_{ij}^{\intercal} x + a_{ij}) \right) \left(\sum_{i=1}^{T} \gamma_i \prod_{j=1}^{P_i} \varphi(u_{ij}^{\intercal} x + c_{ij}) \right) =$$

$$= \sum_{i_1=1}^{K} \sum_{i_2=1}^{T} (\alpha_{i_1}\gamma_{i_2}) \prod_{j_1=1}^{L_{i_1}} \prod_{j_2=1}^{P_{i_2}} \varphi(v_{i_1 j_1}^{\intercal} x + a_{i_1 j_1}) \cdot \varphi(u_{i_2 j_2}^{\intercal} x + c_{i_2 j_2})$$

$$= \sum_{k=1}^{K \cdot T} \beta_{k-1}' \prod_{n=1}^{L_{i_1} + P_{i_2}} \varphi(w_{(k-1)n}^{\intercal} x + b_{(k-1)n})$$

donde:

$$\beta_{k-1}' = \alpha_{i_1}\gamma_{i_2}, \qquad i_1 = (k-1) - T \cdot \lfloor (k-1)/T \rfloor + 1, \qquad i_2 = \lfloor (k-1)/T \rfloor + 1$$

y

$$w_{(k-1)n}^{\intercal} x + b_{(k-1)n} = \begin{cases} v_{i_1 n}^{\intercal} x + a_{i_1 n} & , 1 \leq n \leq L_{i_1} \\ u_{i_2 (n - L_{i_1})}^{\intercal} x + c_{i_2 (n - L_{i_1})} & , L_{i_1} + 1 \leq n \leq L_{i_1} + P_{i_2} \end{cases}$$

Para comprender las fórmulas asociadas a i_1 e i_2, se puede realiza un arreglo matricial de valores representado por (valor del índice i_2, valor del índice i_1), teniendo K columnas y T filas (KT valores). Luego, se numeran estos elementos de izquierda a derecha y arriba hacia abajo, empezando en 0 y terminando en $KT-1$ (ese índice corresponde a $k-1$). Se ajustan los índices a través de traslados para coincidir con los valores correspondientes de i_1 (representando la columna) e i_2 (representando la fila).

Además, dado que la suma de funciones continuas es continua, el producto de funciones continuas es una función continua, y la multiplicación por un escalar de una función continua sigue siendo una función continua. Por lo que $\mathcal{U}$ es un álgebra de funciones reales y continuas sobre I_n. Ahora, se debe verificar que satisface las condiciones de hipótesis del Teorema de Stone - Weierstrass.

- $\mathcal{U}$ separa puntos sobre I_n.

  Sea $x, y \in I_n$ de tal forma que $x \neq y$. Es necesario encontrar un $G \in \mathcal{U}$ que cumple con $G(x) \neq G(y)$.

  Dado que φ es una función no constante, entonces es posible encontrar $a, c \in \mathbb{R}$ tal que $a \neq c$ y $\varphi(a) \neq \varphi(c)$. Ahora se escoge un punto x y un punto y sobre los hiperplanos $\mathcal{P}_1 : \{w^\intercal x + b = a\}$ y $\mathcal{P}_2 : \{w^\intercal x + b = c\}$ y si se toma $G(u) = \varphi(w^\intercal u + b)$ podemos ver que esta función separa puntos por lo siguiente:

  $$G(x) = \varphi(w^\intercal x + b) = \varphi(a)$$
  $$G(y) = \varphi(w^\intercal y + b) = \varphi(c)$$

  por lo que:

  $$G(x) \neq G(y)$$

- $\mathcal{U}$ contiene constantes distintas de 0.

  Sea b del tal forma que $\varphi(b) \neq 0$ y se escoge el vector $w^\intercal = (0, \ldots, 0) \in \mathbb{R}^n$, entonces:

  $$G(x) = \varphi(w^\intercal x + b) = \varphi(b) \neq 0$$

  lo cual es una constante distinta de 0. Si se multiplica por un valor real $\lambda \neq 0$ entonces tenemos que $\mathcal{U}$ contiene todas sus constantes distintas de 0. Al aplicar el teorema de Stone - Weierstrass (7.2.1)tenemos que $\mathcal{U}$ es denso sobre $\mathcal{C}(I_n)$ (Calin, 2020), (Hornik y cols., 1989).

□

NOTA: Es posible generalizar este resultado a cualquier compacto $K \subset \mathbb{R}^n$. Esto porque el teorema 7.2.1 es aplicable para cualquier conjunto compacto de $\mathbb{R}^n$. Si se sustituye I_n por un subconjunto compacto K de $\mathbb{R}^n$, no afecta el resultado del teorema (Calin, 2020).

CAPÍTULO 8

Aplicación del Teorema de Aproximación Universal

Antes de avanzar con la aplicación práctica de los teoremas, primero es importante resaltar que el resultado del teorema 7.4.1 es aplicable al caso particular de una red con una sola capa oculta. Esto por lo siguiente, dado que el producto se cumple para una cantidad finita de N_i, en particular se cumple para $N_i = 1$ para $1 \leq i \leq M$. De esto, se deduce que las sumas finitas de productos finitos de la forma:

$$G(x) = \sum_{i=1}^{M} \beta_i \prod_{j=1}^{N_i} \varphi(w_{ij}^{\intercal} x + b_{ij})$$

pueden expresarse en su caso particular como:

$$G(x) = \sum_{i=1}^{M} \beta_i \varphi(w_i^{\intercal} x + b_i) \tag{1}$$

que tiene la misma forma que las sumas finitas del teorema 6.3.2. Dicho esto, se puede considerar entonces el teorema 7.4.1 como una extensión al Teorema demostrado por Cybenko, pero con la peculiaridad que las funciones de activación no tienen que ser necesariamente funciones sigmoide y que el espacio sobre el cual se trabaja no necesariamente debe ser $I_n = [0, 1]^n$, sino cualquier subconjunto compacto de $\mathbb{R}^n$ (Calin, 2020). Antes de proceder al problema práctico hablaremos un poco de dos tipos de redes que serán de utilidad para la aplicación:

- Red neuronal de propagación hacia adelante (FNN)
- Red neuronal densa (DNN)

8.1. Redes neuronales de propagación hacia adelante (FNN)

Este tipo de redes recibe su nombre debido a que las capas consecutivas pasan información una a la otra desde la capa de entrada hasta la capa de salida. Por lo general, estas redes neuronales son multicapa, es decir, cuentan con más de una sola capa oculta (Bengio, Goodfellow, y Courville,

2015), (Aggarwal, 2018). La estructura general de este tipo de redes supone que todos los nodos de una capa están conectados a todos los nodos de la siguiente capa, sin embargo, no es esencial que esto se cumpla, ya que una FNN puede tener menos conexiones. No obstante, la cantidad de capas ocultas no determina el tipo red, es decir, podemos considerar una red neuronal de una sola capa oculta como una FNN, al igual que una red de N capas ocultas. Lo importante en este tipo de redes es la forma de transmitir la información de una capa a la otra (Aggarwal, 2018). Además, es importante ver que esta estructura de red es la más sencilla que se puede generar partiendo de las capas básicas (entrada, oculta y salida), ya que solo se necesita una capa de entrada, una capa oculta como mínimo y una capa de salida. Por esta razón, a este tipo de red se le considera la red neuronal más básica. Cabe resaltar, que en estos casos también se puede tener una capa de salida multidimensional o unidimensional, pero esto depende intrínsecamente del problema que se quiera resolver. Por ejemplo, para un problema de clasificación binaria nos basta tener un solo nodo en la capa de salida, pero si el problema de clasificación tiene m clases, entonces se requieren m neuronas en la capa de salida (Aggarwal, 2018).

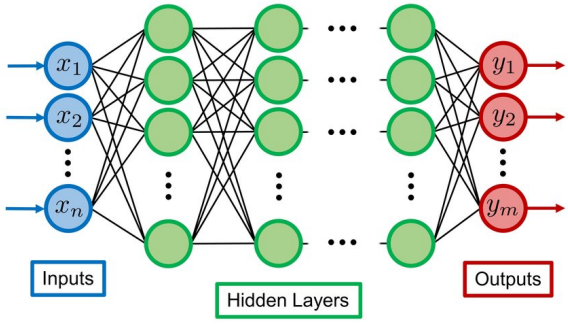

Figura 8.1: Estructura de una FNN.

(Kelly, Lupini, y Epureanu, 2021)

8.2. Redes neuronales densas

Este tipo de redes es bastante peculiar, ya que se le conoce como red neuronal densa a la que todos los nodos de una capa se conectan en su totalidad con los nodos de las capas contiguas. Debido a esto, a este tipo de redes se les conoce también como las redes de capas completamente conectadas y sus capas reciben el nombre de capas densas o capas totalmente conectadas (Aggarwal, 2018). El funcionamiento de dichas capas es exactamente el mismo al de una red neuronal de propagación hacia adelante, ya que la estructura de la red es totalmente similar. Sin embargo, en la mayoría de los casos las redes densas tienden a utilizar múltiples capas densas para incrementar su capacidad computacional (Aggarwal, 2018). No obstante, podemos pensar en la red neuronal de una sola capa oculta como la forma más simple de una red neuronal densa. Esto sucede porque la neurona de entrada, está completamente conectada a todos los nodos de la capa oculta, y los nodos de la capa oculta están completamente conectados a la capa de salida. Dicho esto, podemos pensar que los teoremas de aproximación universal presentados en los capítulos 6 y 7 que cumplen con la forma de la ecuación 1 son criterios aplicables para las redes densas de una sola capa oculta siempre y cuando se ingrese un vector como entrada de la red y se obtenga un vector en la salida de la red (Bengio y cols., 2015) (Nielsen, 2018). Para garantizar que una capa es densa, basta con revisar la cantidad de conexiones de entrada y la cantidad de conexiones de salida de cada una de las capas. Para ello, suponga que se tiene un red neuronal con una capa de entrada que tiene m valores de entrada, una capa oculta con una función de activación continua y no constante f con n nodos, y una capa de salida con r número de clases, donde $m \geq 1, n \geq 1, r \geq 1$ con $m, n, r \in \mathbb{Z}^+$. Entonces, para compro-

bar que la red neuronal es densa, es necesario que se cumpla que entre la capa de entrada y la capa oculta hay $m \times n$ conexiones y $m \times n + n$ parámetros, y, entre la capa oculta y la capa de salida debe haber $n \times r$ conexiones o parámetros. Es necesario recalcar que el total de parámetros es la suma de conexiones entre las capas (determinando la cantidad de pesos) y la cantidad de neuronas en la capa oculta (determinando la cantidad de sesgos). Además, bajo este modelo, se supone que la capa de salida no tiene sesgos, por ello la cantidad de conexiones y la cantidad de parámetros coinciden. En la siguiente ilustración se puede observar más a detalle esta relación, cabe resaltar que se supone que $n > m \geq r$ (Aggarwal, 2018).

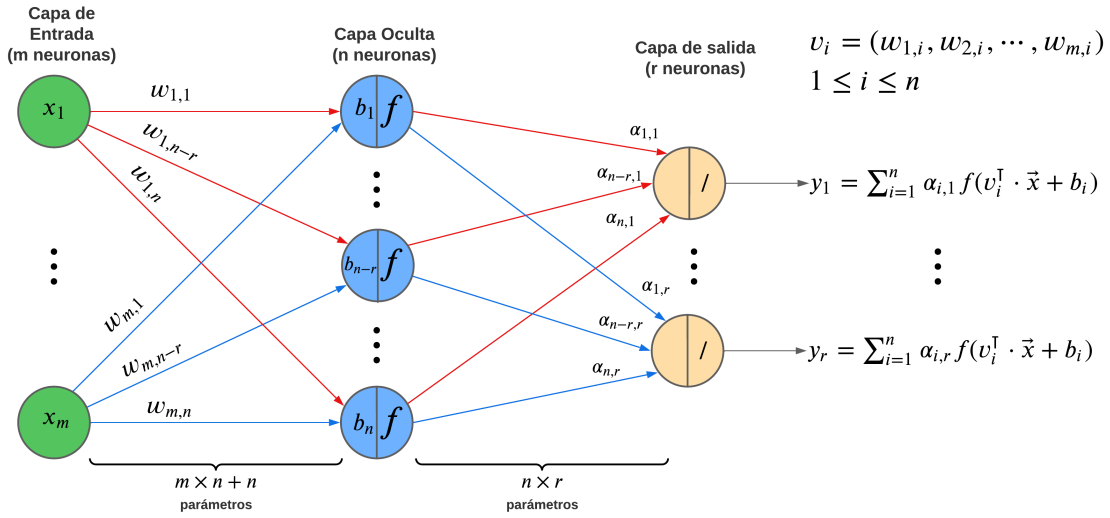

Figura 8.2: Ilustración de una Red Neuronal Densa de una sola capa oculta generalizada.

8.3. El lenguaje de señas guatemalteco

Es necesario recalcar que existe un lenguaje de señas universal. Sin embargo, cada país tiene el derecho de establecer su propio lenguaje de señas. Tal es el caso de Guatemala, que desde 2020 por medio del Decreto 3 - 2020 se garantizó LENSEGUA como la Lengua de señas oficial de Guatemala (LENSEGUA, 2023). El alfabeto se conforma por 27 letras, cada una con su respectiva traducción al alfabeto español. No obstante, es importante recalcar lo siguiente, algunas letras cuentan con movimiento dentro del alfabeto de señas. Por ello, en este trabajo se trabajará con una versión estática de dichas letras. Las letras que cuentan con movimiento son tres en total: F, J y S. Además, es necesario resaltar que algunas letras solo utilizan una mano, mientras que otras, como la F, la Ñ, la Q y la X, requieren el uso de ambas manos. Esto repercute en la forma de trabajar el modelo más adelante. En adición a ello, es conocimiento crítico que al momento de realizar este trabajo no se cuenta con una base de datos de imágenes del alfabeto de LENSEGUA. Por ello, parte de la aplicación conlleva la recopilación y procesamiento de las mismas con la finalidad de adaptar el problema de tal forma que sea posible emplear los resultados obtenidos en los capítulos 6 y 7. Entonces, podemos desglosar el desarrollo del modelo en cinco partes:

1. ¿Por qué es aplicable el teorema?
2. Recopilación de los datos
3. Procesamiento de los datos

4. Entrenamientos del modelo
5. Emplear el modelo para predicción en tiempo real

Figura 8.3: Alfabeto de LENSEGUA.

(ProCiegos, 2023)

8.4. Desarrollo del modelo

Para la implementación práctica se hizo uso de dos modelos de Redes Neuronales de una sola capa oculta, cada uno entrenada por separado debido a su naturaleza. La razón para trabajar con dos modelos es sencilla, las letras que utilizan ambas manos requieren el doble de variables de entrada en comparación con las que solo utilizan una mano. Por ello, fue necesario trabajarlos de manera distinta en las etapas de recopilación, preparación de datos y el entrenamiento. Todas las etapas del desarrollo se llevaron a cabo en el lenguaje de programación Python. En las siguientes secciones nos referiremos a "Modelo 1", como el modelo que trabajó las letras que utilizan una sola mano y a "Modelo 2", como el que trabajó las letras que utilizan ambas manos.

8.4.1. ¿Por qué es aplicable el teorema?

Antes de entrar a detalle a la parte aplicada, es importante establecer el porqué los resultados obtenidos en los capítulos 6 y 7 son aplicables a este problema. La forma en la que se busca abordar la solución es mediante la detección puntos importantes en cada una de las manos, y luego extraer las coordenadas (x, y) de dichos puntos. De esto se tiene que por cada punto mapeado en una mano se tienen dos coordenadas que se utilizaran como datos de entrada para el modelo. Por lo tanto, si se identifican m puntos en una mano, eso quiere decir que tendremos un vector de entrada de $\vec{x}_1 \in \mathbb{R}^{2m} \subset \mathbb{R}^n$ para el caso de una sola mano, mientras que para el caso de ambas manos se

tendrá un vector $\vec{x}_2 \in \mathbb{R}^{4m} \subset \mathbb{R}^n$. Además, es necesario recalcar que estas imágenes fueron tomadas por medio de una cámara, la cual se puede interpretar como un conjunto acotado y cerrado (una cámara tiene una dimensión finita de pixeles y contiene su borde), en el contexto de $\mathbb{R}^n$ esto es una caracterización de un conjunto compacto. Por lo tanto, las imágenes generadas por medio de la cámara son un subconjunto compacto de $\mathbb{R}^n$. Por ende, los teoremas son aplicables siempre y cuando se utilicen funciones continuas no constantes y un modelo de red de una sola capa oculta. Además, dado que los vectores $\vec{x}_1$ y $\vec{x}_2$ determinan la cantidad de neuronas de entrada, y el total de letras del alfabeto determina la cantidad de neuronas de salida para el modelo, lo único que hace falta determinar es la función de activación a utilizar y la cantidad de nodos que conforman la capa oculta.

8.4.2. Recopilación de datos

Para este proceso se programó un código que se conecta directamente a cualquier cámara vinculada a la computadora (en este caso la cámara integrada en la Laptop), por medio del lenguaje de programación Python y una librería especial de visión computacional llamada OpenCV (OpenCV, 2023). Esto permite, por medio de comandos, tomar fotografías en tiempo real desde el módulo de cámara indicado, con un cierto intervalo de tiempo entre cada toma. En este proyecto las capturas se tomaron cada segundo. Dicho esto, se hizo un mayor número de muestras para las letras que utilizan dos manos, dado que durante el procesamiento se descartaban muchas más imágenes. Cabe resaltar que el muestreó consistió en tomar la posición de lenguaje de señas de una letra y realizarla frente a la cámara, durante todo el tiempo de recopilación. Además, procurando mover la mano en todas las direcciones posibles (arriba, abajo, izquierda, derecha, hacia adelante y hacia atrás) con la finalidad de conseguir la mayor cantidad de información.

Modelo	Código de letra	Letra	Cantidad de muestras
1	0	A	200
1	1	B	200
1	2	C	200
1	3	D	200
1	4	E	200
1	5	G	200
1	6	H	200
1	7	I	200
1	8	J	200
1	9	K	200
1	10	L	200
1	11	M	200
1	12	N	200
1	13	O	200
1	14	P	200
1	15	R	200
1	16	S	200
1	17	T	200
1	18	U	200
1	19	V	200
1	20	W	200
1	21	Y	200
1	22	Z	200
	Subtotal	23 letras	4,600 imágenes
2	0	F	400
2	1	Ñ	400
2	2	Q	400
2	3	X	400
	Subtotal	4 letras	1,600 imágenes
	Total	27 letras	6,200 imágenes

Tabla 8.1: Datos recopilados.

Figura 8.4: Muestra de la letra L.

Figura 8.5: Muestra de la letra Q.

8.4.3. Procesamiento de los datos

Una vez recopilada la información necesaria, se procedió con el procesamiento de las imágenes. Para ello, empleó la librería MediaPipe y el ya mencionado OpenCV de Python (Google, 2023), (OpenCV, 2023). Esta librería fue desarrollada por Google con el objetivo de poder facilitar la detección de objetos, en particular, la librería puede detectar manos y dedos. Primero se cargan las imágenes recopiladas junto con su etiqueta, la letra a la que corresponde. Luego, con ayuda de OpenCV se transformaron las imágenes de un formato BGR a formato RGB. Esto con la finalidad de trabajar más cómodamente dentro de la librería, dado que la mayoría de librerías de visión computacional funcionan mejor en formato RGB. Posteriormente, con ayuda de la librería MediaPipe Hands, se buscó las manos en cada imagen, sin embargo, no en todas las imágenes fue posible encontrar una mano (Google, 2023), (OpenCV, 2023). Debido a ello, en este proceso de identificación se descartaron bastantes imágenes en ambos modelos. Sobretodo, en el Modelo 2, en dónde se fue más estricto aún, dado que debía encontrarse como mínimo dos manos en cada imagen. Esto se puede observar más a detalle en el cuadro 8.2 donde se explora con mayor profundidad la cantidad de datos procesados adecuadamente para cada clase. El mal procesamiento puede deberse a varios factores, como la calidad de la cámara, la posición de la mano en ese momento, si la imagen está mal enfocada, la luz, entre otros.

Posteriormente, una vez reconocidas las manos, es posible mapear puntos específicos dentro de la imagen donde se encuentran las coyunturas de los dedos de cada una. En total, por cada imagen procesada de forma adecuada es posible identificar veintiún puntos importantes que conforman la mano completa. Para el caso de las letras que utilizan dos manos, este número se duplica a cuarenta y dos puntos por cada muestra. Estos puntos, juegan un papel importante dentro de la aplicación, debido a que, es posible extraer las coordenadas que los conforman dentro de la imagen. Si bien la librería permite extraer coordenadas en los tres ejes, en este proyecto únicamente se extrajo las coordenadas planas (x, y) de cada punto (Google, 2023). Al tener dos coordenadas por cada punto, entonces se generó un vector compuesto por las cuarenta y dos coordenadas en el caso de una sola mano, y un vector de ochenta y cuatro coordenadas para las letras que utilizan ambas manos. Estas coordenadas fueron empaquetadas junto con la letra del alfabeto que representan, esto con la finalidad de hacer uso de los vectores generados para posteriormente entrenar los modelos con dicha información.

Modelo	Letra	Datos procesados adecuadamente
1	A	149
1	B	184
1	C	116
1	D	200
1	E	172
1	G	135
1	H	200
1	I	133
1	J	200
1	K	200
1	L	140
1	M	200
1	N	169
1	O	200
1	P	195
1	R	200
1	S	200
1	T	200
1	U	46
1	V	111
1	W	84
1	Y	200
1	Z	200
Subtotal	23 letras	3,834 imágenes
2	F	216
2	Ñ	126
2	Q	98
2	X	317
Subtotal	4 letras	747 imágenes
Total	27 letras	4,581 imágenes

Tabla 8.2: Datos procesados adecuadamente.

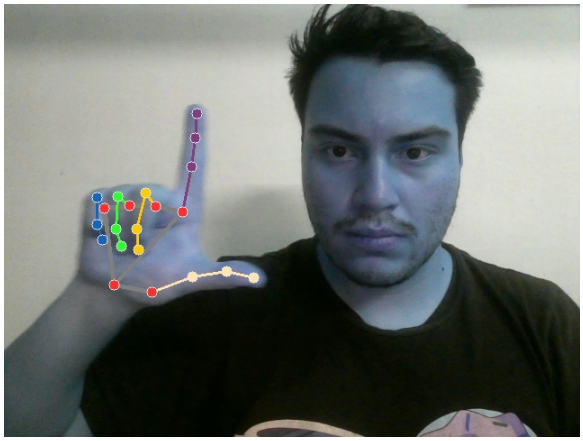

Figura 8.6: Muestra de la letra L procesada.

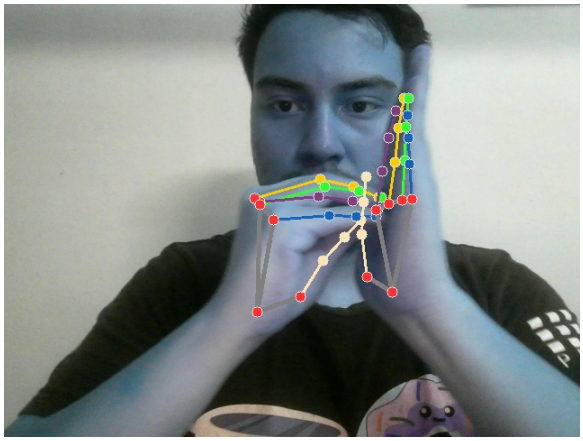

Figura 8.7: Muestra de la letra Q procesada.

8.4.4. Entrenamiento del modelo

Ahora, por el procesamiento de los datos se tiene dos vectores de coordenadas $\vec{x}_1 \in \mathbb{R}^{42}$ y $\vec{x}_2 \in \mathbb{R}^{84}$ para las imágenes que utilizan una sola mano y ambas manos, respectivamente. Esto nos permite proponer la estructura de una red neuronal para cada uno de los modelos en cuestión. Cabe resaltar, que la cantidad de nodos utilizados en la capa oculta y la función de activación fue determinada por el autor de este trabajo, no obstante, es probable que existan otras funciones de activación o cantidad de nodos que también brinden buenos resultados tanto en la etapa de entrenamiento como en las predicciones en tiempo real. A continuación, se muestra las características utilizados para la creación de los modelos respectivos:

Modelo	**Nodos de entrada**	**Capa oculta**	**Nodos de salida**	**Función**
1	42	42	23	Sigmoide, $c = 1$
2	84	10	4	ReLU

Tabla 8.3: Características utilizadas.

Por lo que podemos expresar los modelos matemáticamente de la siguiente manera:

- Modelo 1:

$$y_j = \sum_{i=1}^{42} \alpha_{ij} \sigma(v_i^\intercal \cdot \vec{x}_1 + b_i)$$

donde y_j representa las entradas del vector $\vec{y}_1 \in \mathbb{R}^{23}$, $v_i \in \mathbb{R}^{42}$, y $b_i, \alpha_{ij} \in \mathbb{R}$, para $j = 1, \ldots, 23$, respectivamente.

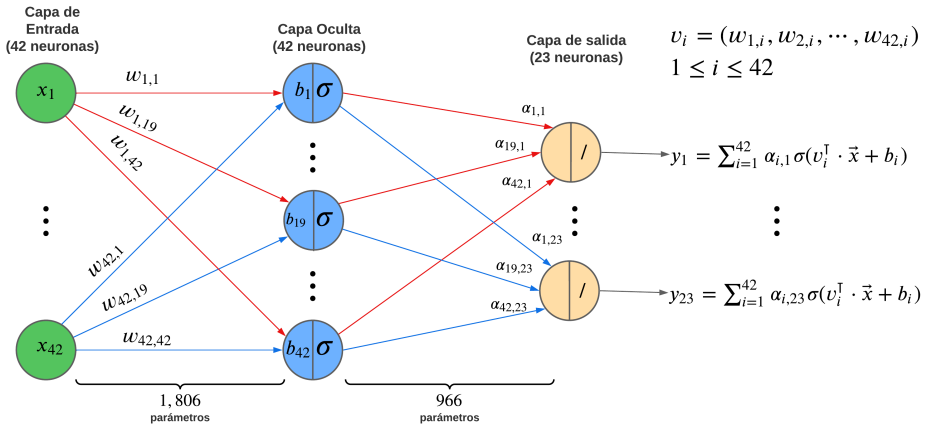

Figura 8.8: Estructura del Modelo 1.

- Modelo 2:

$$y_k = \sum_{i=1}^{10} \alpha_{ik} ReLU(v_i^\intercal \cdot \vec{x}_2 + b_i)$$

donde y_k representa las entradas del vector $\vec{y}_2 \in \mathbb{R}^4$, $v_i \in \mathbb{R}^{84}$, y $b_i, \alpha_{ik} \in \mathbb{R}$ para $k = 1, 2, 3, 4$, respectivamente.

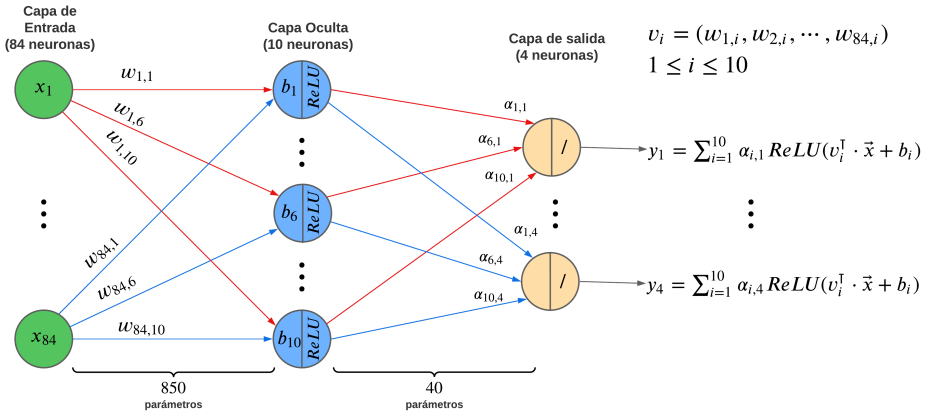

Figura 8.9: Estructura del Modelo 2.

Posteriormente, se programó el entrenamiento de los modelos anteriormente mencionados. Con la ayuda de la librería Keras, es posible armar una Red Neuronal con las especificaciones indicadas anteriormente, además, para la fase de entrenamiento se optó por utilizar el 80 % de los datos, mientras que para la etapa de validación se utilizó el restante 20 %, para cada modelo respectivamente (Keras, 2023).

Modelo	Datos de entrenamiento	Datos de prueba	Accuracy
1	3,068	766	99.34 %
2	598	149	99.81 %

Tabla 8.4: Resultados de entrenamiento y prueba.

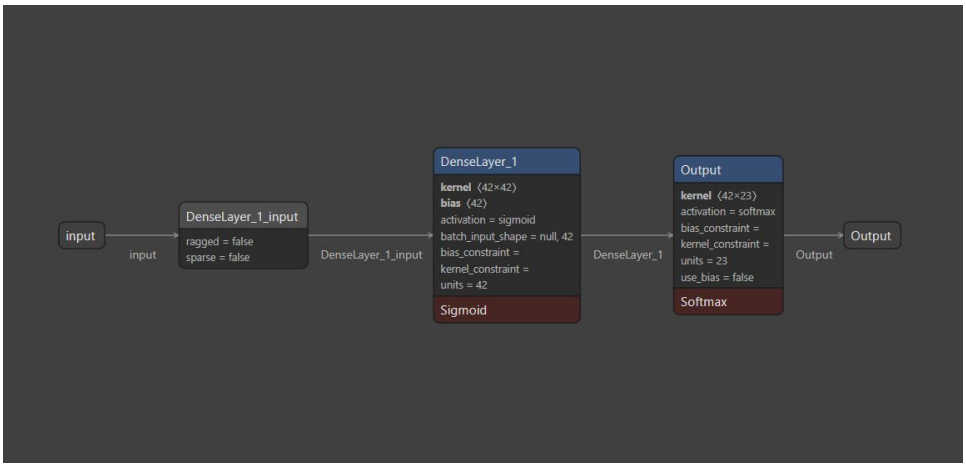

Figura 8.10: Estructura del Modelo 1.

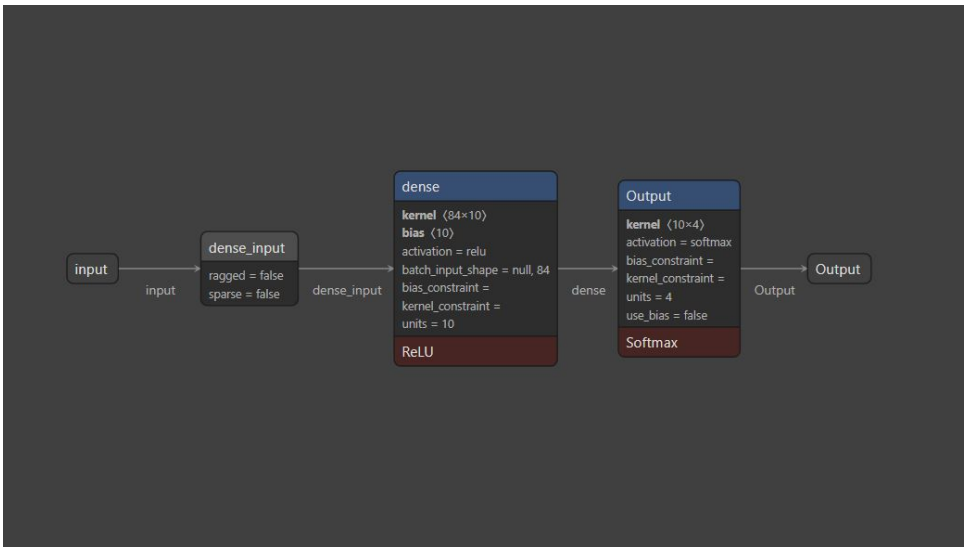

Figura 8.11: Estructura del Modelo 2.

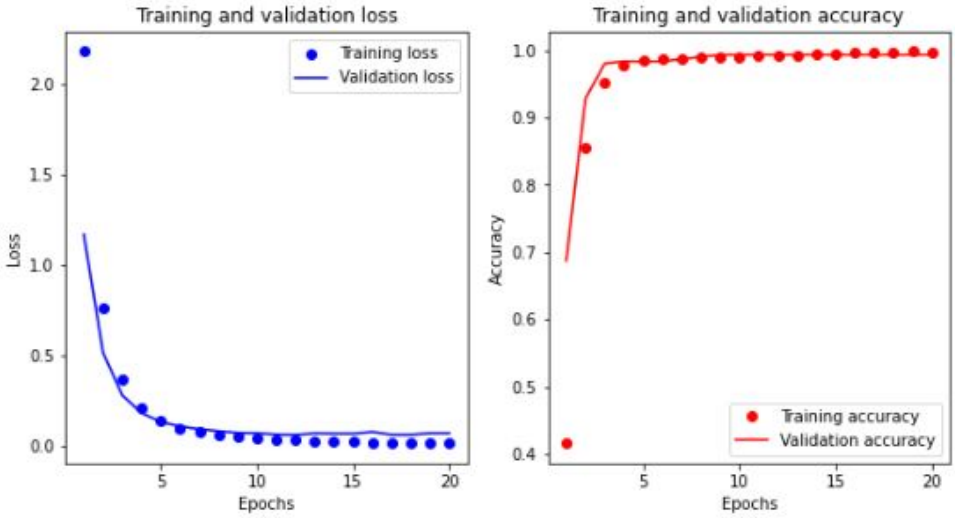

Figura 8.12: Resultados del entrenamiento de Modelo 1.

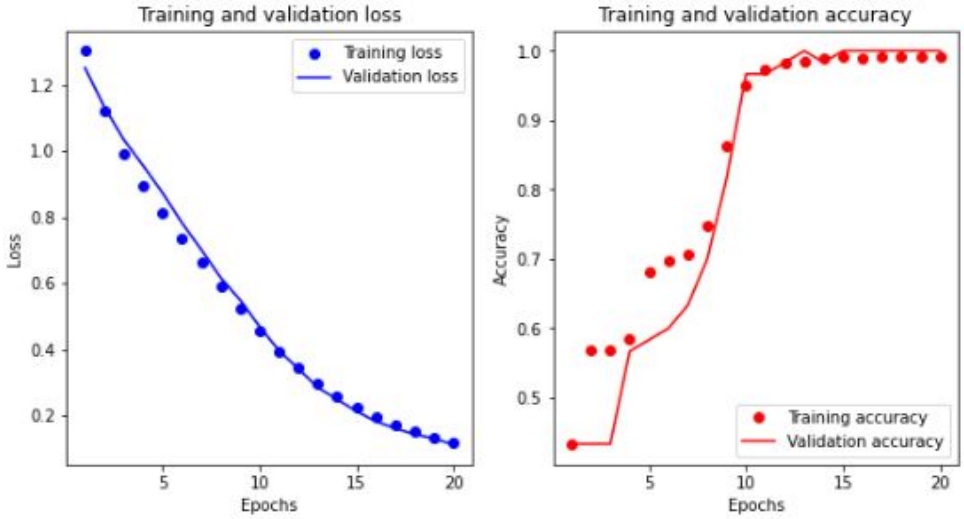

Figura 8.13: Resultados del entrenamiento de Modelo 2.

8.4.5. Ejecución en tiempo real

Para esta parte, se emplearon los modelos anteriormente entrenados y empaquetados, esto con la finalidad de cargarlos y ejecutarlos en una aplicación que se conecta directamente a un módulo de cámara. Esto es posible gracias a la librería OpenCV que se conecta directamente a una cámara en específico, luego se procesan los datos captados por la cámara en tiempo real con la ayuda de la librería MediaPipe (Google, 2023), (OpenCV, 2023). Finalmente, los datos procesados ingresan a uno de los dos modelos. Si el programa detecta que se está empleando una sola mano, entonces los parámetros de entrada pasan al Modelo 1. En caso contrario, si se detectan dos manos, los parámetros de entrada pasan al Modelo 2. La aplicación se encarga de ejecutar las predicciones en tiempo real y desplegarlas en la pantalla, junto con su respectiva probabilidad de predicción. Estas lecturas se llevan a cabo a una velocidad de treinta cuadros por segundo. A continuación se presentan dos ejemplos del uso de la aplicación en tiempo real, y sus respectivos resultados:

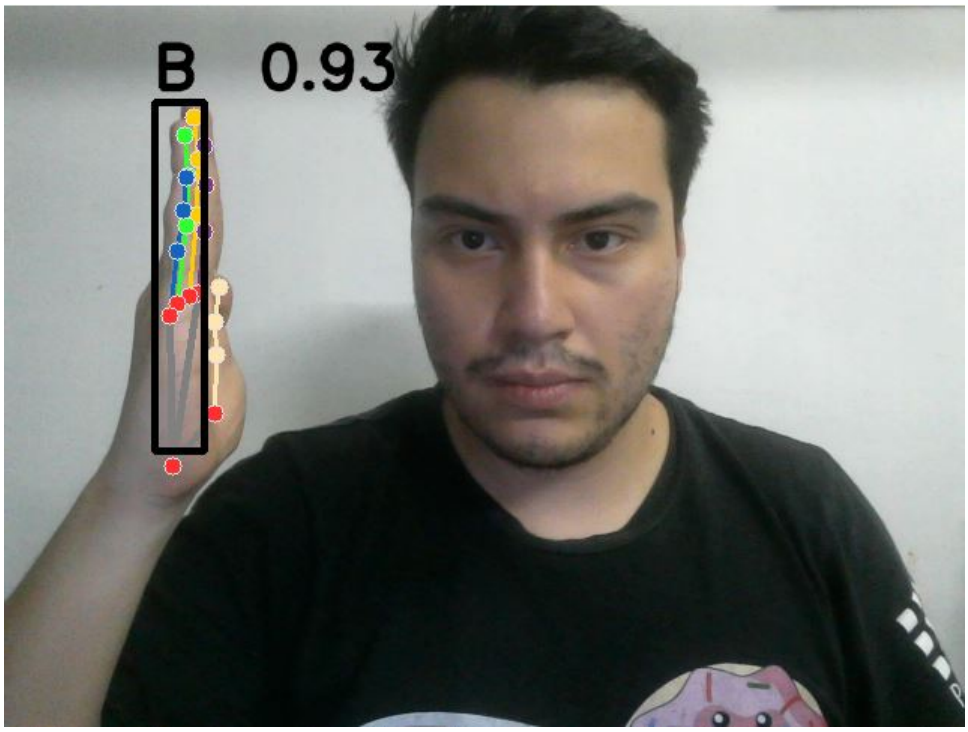

Figura 8.14: Predicción de la letra B.

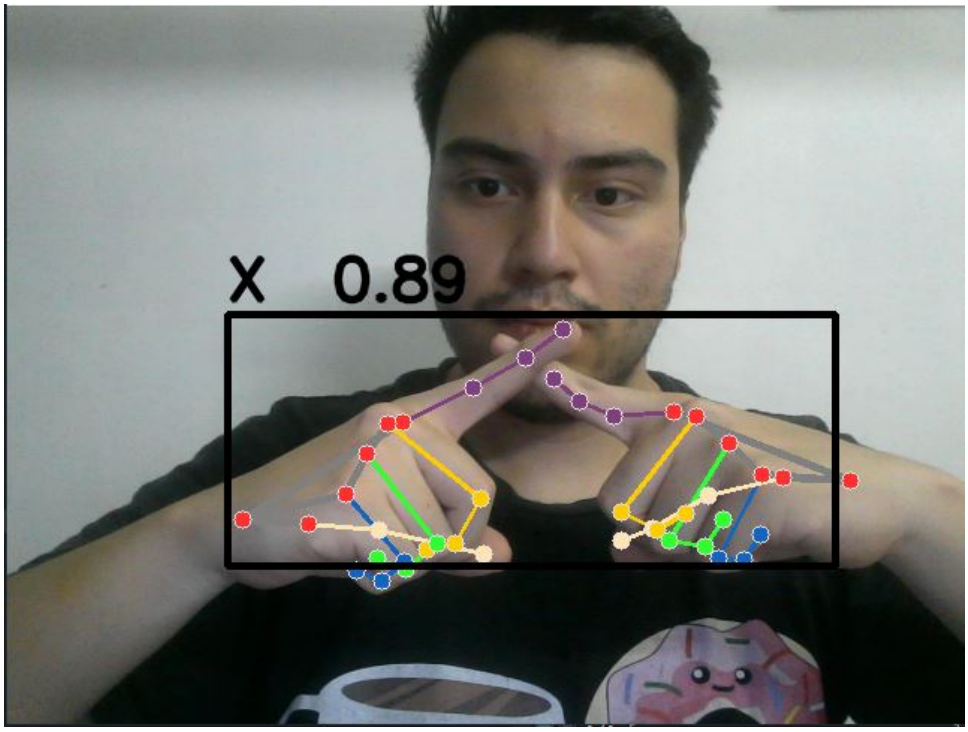

Figura 8.15: Predicción de la letra X.

CAPÍTULO 9

Conclusiones

1. Se cumplió con el primer objetivo específico, dado que fue posible presentar todas las herramientas (Teorema de Convergencia Dominada de Lebesgue, Teorema de Representación de Riesz, Teorema de Hahn - Banach y Teorema de Stone - Weierstrass) y ramas de la Matemática necesarias (Análisis de Variable Real, Teoría de la Medida y Análisis Funcional) para exponer las dos versiones del Teorema de Aproximación Universal.

2. El segundo objetivo específico se cumplió porque se incluyeron los detalles de las demostraciones para tener una versión de los teoremas más adecuada al lector, que facilita su lectura y compresión en comparación con la bibliografía consultada.

3. Las Redes Neuronales implementadas comprobaron ser una buena solución del problema, dado que en la etapa de entrenamiento se obtuvo una alta exactitud en ambos modelos. Además, en la aplicación práctica, durante la etapa de prueba, es posible observar que este alto grado de exactitud se mantiene, aún analizando datos en tiempo real. Por lo que, los modelos implementados logran reconocer de manera correcta las letras del alfabeto de LENSEGUA, lo que representa un buen modelo de clasificación, lo que cumple con el tercer objetivo específico.

4. Cybenko fue la primera persona en probar el Teorema de Aproximación Universal para el caso de N nodos en la capa oculta, para las redes $\sum$ - clase. Sin embargo, su versión del teorema está limitada a las funciones de activación sigmoide, sobre el hipercubo $I_n = [0,1]^n$. No obstante, al generalizar este teorema para cualquier función de activación continua no constante, fue necesario integrar el concepto de red $\sum \prod$ - clase.

5. Si bien los resultados obtenidos en este trabajo son aplicables para las Redes Neuronales más básicas, estos pueden utilizarse de manera versátil para resolver problemas de alta complejidad siempre y cuando se realice un procesamiento de datos adecuado. Además, esto permite apreciar la alta capacidad y flexibilidad que tienen estos modelos, lo que los convierte en herramientas útiles para distintas aplicaciones.

CAPÍTULO 10

Recomendaciones

1. El resultado del Teorema de Aproximación Universal obtenido en este trabajo tiene bastantes restricciones en cuanta la cantidad de capas y el tipo de red que puede utilizarse. Por ello, se recomienda hacer investigación sobre las extensiones de este para los casos de una Red Neuronal Multicapa y el caso de Redes Neuronales Convolucionales.

2. Para la recopilación de datos se recomienda tomar en consideración algoritmos de análisis de video, con la finalidad de poder integrar los movimientos a las letras del alfabeto que así lo requieren. Ya que en este trabajo se utilizó la versión estática de cada una de ellas. Para esto puede abocarse a instituciones como ASORGUA o el Comité de Pro Ciegos y Sordos de Guatemala, quienes podrían estar interesados en colaborar con la recopilación de datos.

3. Se recomienda emplear otras técnicas de procesamiento de imágenes, tales cómo detección de rostro y detección de postura para ampliar la gama de gestos y señas que los modelos pueden aprender y reconocer. Si bien la aplicación práctica presentada en este trabajo fue bastante acertada, es necesario considerar que el lenguaje de señas es mucho más complejo y versátil de lo que un alfabeto puede albergar.

4. Para profundizar más en el tema es aconsejable tener un conocimiento básico de Análisis de Variable Real, Análisis Funcional, Cálculo, Topología, Teoría de Conjuntos y sobre todo, Teoría de la Medida. Este último es bastante importante, ya que las generalizaciones del teorema utilizan en gran cantidad definiciones de esta rama de la Matemática.

5. Antes de pensar en abordar el tema, es importante comprender cómo se ha abordado convencionalmente. Se recomienda investigar la literatura existente y los estudios previos relacionados al tópico en cuestión. Esto formará una base sólida para entender a profundidad los enfoques utilizados y las limitaciones relacionadas al tema.

6. El Teorema de Aproximación Universal establece las capacidades teóricas de las Redes Neuronales. Sin embargo, no provee una guía de cómo construir la red, determinar la arquitectura óptima o el entrenamiento de los datos. Por lo tanto, se recomienda investigar más estas estrategias prácticas para utilizar el teorema de forma correcta.

CAPÍTULO 11

Bibliografía

Referencias

Aggarwal, C. (2018). *Neural networks and deep learning.* New York, USA: Springer verlag.

Bengio, Y., Goodfellow, I., y Courville, A. (2015). *Deep learning.* Boston: MIT Press.

Bernués, J. (2010). El teorema de stone - weierstrass. *La Gaceta de la RSME*, *13*, 705-711.

Calin, O. (2020). *Deep learning architectures: A mathematical approach.* Switzerland: Springer Verlag.

Chen, J. (2023). *What is a neural network?* Descargado de `https://www.investopedia.com/terms/n/neuralnetwork.asp#:~:text=What%20Are%20the%20Components%20of,layer%2C%20and%20an%20output%20layer.`

Clabaugh, C., Myzewski, D., y Pang, J. (2023). *Neural networks - history.* Descargado de `https://cs.stanford.edu/people/eroberts/courses/soco/projects/neural-networks/Sources/index.html`

College, D. (2023). *George cybenko.* Descargado de `https://sites.dartmouth.edu/cybenko/`

Cybenko, G. (1989). Approximation by superposition of a sigmoidal function. *Mathematics of Control, Signals, and Systems*(1), 303-313.

De Barra, G. (2003). *Measure theory and integration.* Philadelphia: Woodhead Publishing.

Google. (2023). *On-device machine learning for everyone.* Descargado de `https://developers.google.com/mediapipe`

Hornik, K., Stinchcombe, M., y White, H. (1989). Multilayer feedforward networks are universal aproximators. *Neural Networks*(2), 359-366.

Kelly, S., Lupini, A., y Epureanu, B. (2021, 04). Data-driven modeling approach for mistuned cyclic structures. *AIAA Journal*, *59*, 1-13. doi: 10.2514/1.J060117

Keras. (2023). *About keras.* Descargado de `https://keras.io/about/`

Kolmogorov, A., y Formin, S. (1954). *Elements of the theory of functions and functional analysis.* New York: Graylock Press.

LENSEGUA. (2023). *¿qué es lensegua?* Descargado de `https://lensegua.com/`

Masher, M., y D'Bannon, M. (2016). *A concise history of neural networks.* Descargado de `https://towardsdatascience.com/a-concise-history-of-neural-networks-2070655d3fec`

McNeela, D. (2017). *The universal aproximation theorem for neural networks.* Descargado de `https://mcneela.github.io/machine_learning/2017/03/21/Universal-Approximation-Theorem.html#menu`

Nielsen, M. A. (2018). *Neural networks and deep learning* [misc]. Determination Press. Descargado de `http://neuralnetworksanddeeplearning.com/`

OpenCV. (2023). *Generative ai is the next big thing. master it.* Descargado de https://opencv.org/

ProCiegos. (2023). *Alfabeto manual guatemalteco.* Descargado de https://www.prociegosysordos.org.gt/alfabetos.html

Royden, H. (1988). *Real analysis.* New York: Macmillan Publishing Company.

Rudin, W. (1953). *Principles of mathematical analysis.* New York: McGraw Hill.

Rudin, W. (1991). *Functional analysis.* New York: Mc-Grawhill, Inc.

Runde, V. (2005). *A taste of topology.* San Francisco: Springer Verlag.

Salamon, D. (2020). *Measure and integration.* Zurich: ETH Zurich.

Sciences, L. (2023). *Prof. ding-xuan zhou.* Descargado de https://www.cityu.edu.hk/rcms/DXZhou.htm

CAPÍTULO 12

Anexos

12.1. Teoría de Conjuntos

Definición 12.1.1. ***Función Indicador****: la función indicador o función característica de un subconjunto de un conjunto es una función que mapea elementos del subconjunto en 1, y 0 en cualquier otro caso. Suponga $A \subset X$ entonces f es la función indicador definida como:*

$$1_A = f(x) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$$

12.2. Análisis de Variable Real

Teorema 12.2.1. ***M - Test de Weierstrass****: Suponga que $\{f_n\}$ es una sucesión de funciones definida sobre A, y además:*

$$|f_n(x)| \leq M_n, \qquad \forall n \in \mathbb{N}$$

entonces la serie $\sum f_n$ converge uniformemente sobre A si $\sum M_n$ converge (Rudin, 1953).

Teorema 12.2.2. *La serie binomial generada por*

$$f(x) = (1-x)^{1/2}$$

converge uniformemente sobre el intervalo $[0, 1]$.

Demostración. Al aplicar el Teorema Generalizado del Binomio sobre $f(x) = (1-x)^{1/2}$ obtenemos:

$$f(x) = (1-x)^{1/2} = \sum_{n=0}^{\infty} \binom{1/2}{n} (-1)^n x^n$$

entonces sabemos que:

$$f_n(x) = \binom{1/2}{n} (-1)^n x^n, \qquad \forall x \in [0, 1]$$

Luego, es posible demostrar por inducción los siguientes resultados:

$$\binom{1/2}{n} = \binom{2n}{n} \frac{(-1)^{n+1}}{2^{2n}(2n-1)}$$

$$\binom{2n}{n} \leq \frac{2^{2n}}{\sqrt{n+1}}$$

y por lo tanto tenemos lo siguiente:

$$\left|\binom{1/2}{n}(-1)^n x^n\right| \leq \left|\binom{1/2}{n}\right| = \left|\binom{2n}{n}\frac{(-1)^{n+1}}{2^{2n}(2n-1)}\right| \leq \left|\left(\frac{2^{2n}}{\sqrt{n+1}}\right)\cdot\frac{1}{2^{2n}(2n-1)}\right|, \qquad \forall x \in [0,1]$$

y por ende:

$$|f_n(x)| \leq \frac{1}{\sqrt{n+1}(2n-1)} = M_n, \qquad \forall x \in [0,1], n \geq 1$$

Ahora es necesario demostrar que $\sum M_n$ converge. Para ello se emplea el Criterio de condensación de Cauchy:

$$\sum_{n=0}^{\infty} \frac{2^n}{\sqrt{2^n+1}(2^{n+1}-1)}$$

y al aplicar el Criterio de la Razón se tiene:

$$\lim_{n\to\infty}\left|\frac{2^{n+1}\sqrt{2^n+1}(2^{n+1}-1)}{2^n\sqrt{2^{n+1}+1}(2^{n+2}-1)}\right| = \left|2\cdot\frac{1}{\sqrt{2}}\cdot\frac{1}{2}\right| = \frac{\sqrt{2}}{2} < 1$$

por lo que la serie:

$$\sum_{n=0}^{\infty} \frac{2^n}{\sqrt{2^n+1}(2^{n+1}-1)}, \qquad \text{converge.}$$

y por el Criterio de Condensación también lo hace $\sum M_n$, por lo que se puede concluir por el M - Test que:

$$f(x) = (1-x)^{1/2}, \qquad \text{converge uniformemente sobre } [0,1]$$

□

12.3. Topología

Definición 12.3.1. ***Cubierta Abierta****: Sea (X,τ) un espacio topológico, y sea $S \subset X$. Una cubierta abierta para S es la colección $\mathcal{U}$ de subconjuntos abiertos de X de tal forma que $S \subset \bigcup\{U : U \in \mathcal{U}\}$ (Runde, 2005).*

Definición 12.3.2. ***Compacto****: Un subconjunto K del espacio topológico (X,τ) es compacto si para cada cubierta abierta $\mathcal{U}$ de K hay $U_1,\ldots,U_n \in \mathcal{U}$ de tal forma que $K \subset U_1 \cup \ldots \cup U_n$ (Runde, 2005).*

12.4. Teoría de la Medida

Teorema 12.4.1. ***Teorema de Convergencia Dominada de Lebesgue****: sea $(X,\mathcal{A},\mu)$ un espacio de medida, sea $g : X \to [0,\infty)$ una función integrable y sea $f_n : X \to \mathbb{R}$ una sucesión de funciones integrables que satisfacen:*

$$|f_n(x)| \leq g(x)$$

para todo $x \in X$ y $n \in \mathbb{N}$ y que convergen puntualmente a $f : X \to \mathbb{R}$, es decir:

$$f(x) = \lim_{n\to\infty} f_n(x)$$

para todo $x \in X$. Entonces f es integrable y para cualquier $E \in \mathcal{A}$ se tiene que:

$$\int_E f d\mu = \lim_{n\to\infty}\int_E f_n d\mu$$

(Salamon, 2020)

Definición 12.4.1. *σ - **álgebra de Borel**: la σ - álgebra de Borel se relaciona con la topología de un conjunto de la siguiente manera: sea (X, τ) un espacio topológico y sea $\mathcal{B} \subset 2^X$ la σ- álgebra más pequeña que contiene a τ, entonces $\mathcal{B}$ es el σ- álgebra de Borel de (X, τ). A los elementos de $\mathcal{B}$ se les conoce como conjuntos (medibles) de Borel. Esta σ - álgebra se compone por lo conjuntos abiertos, o cerrados. (Salamon, 2020)*

Definición 12.4.2. ***Medida de Borel**: una medida $\mu : \mathcal{B} \to [0, \infty]$ se llama medida de Borel si $\mu(K) < \infty$ para todo conjunto compacto $K \subset X$. (Salamon, 2020)*

Definición 12.4.3. ***Medida de regular externa de Borel**: una medida de Borel es regular externa si:*

$$\mu(B) = \text{ínf}\{\mu(U) | B \subset U \subset X \text{ y } U \text{ es abierto } \}$$

(Salamon, 2020)

Definición 12.4.4. ***Medida de regular interna de Borel**: una medida de Borel es regular interna si para todo $B \subset \mathcal{B}$ se tiene que :*

$$\mu(B) = \sup\{\mu(K) | K \subset B \text{ y } K \text{ es compacto}\}$$

(Salamon, 2020)

Definición 12.4.5. ***Medida regular de Borel**: una medida de Borel es regular si es regular externa y regular interna. (Salamon, 2020)*

Definición 12.4.6. *$\sigma-$ **compacto**: se dice que X es $\sigma-$compacto si existe un secuencia de conjuntos compactos $K_i \subset X, i \in \mathbb{N}$ de tal forma que $K_i \subset K_{i+1}$ para todo i y $X = \bigcup_{i=1}^{\infty} K_i$ (Salamon, 2020)*

Teorema 12.4.2. *Sobre un σ - compacto la medida regular interna y la medida regular externa de Borel coinciden, es decir, la medida es regular de Borel. (Salamon, 2020)*

Teorema 12.4.3. *$\mathbb{R}^n$ es $\sigma-$compacto. (Salamon, 2020)*

Teorema 12.4.4. *$I_n = [0, 1]^n$ es $\sigma-$compacto. (Salamon, 2020)*

Teorema 12.4.5. *Las funciones continuas son medibles. (De Barra, 2003)*

Teorema 12.4.6. *Sea $\{f_i\}$ una sucesión de funciones medibles que convergen puntualmente a f, entonces f es medible. (De Barra, 2003)*

Teorema 12.4.7. *Si f es una función Riemann Integrable y acotada sobre un intervalo $[a, b]$, entonces f es integrable en le sentido de Lebesgue (las integrales coinciden). (De Barra, 2003)*

Teorema 12.4.8. *Sea E un conjunto medible, y $1 \leq p \leq \infty$. Entonces el subconjunto de funciones simples de $L^p(E)$ es denso en $L^p(E)$. (Royden, 1988)*

Definición 12.4.7. ***Función de Soporte Compacto**: sea (X, U) un espacio compacto de Hausdorf y $\mathcal{B}$ su $\sigma-$ álgebra de Borel. Una función $f : X \to \mathbb{R}$ se llama de soporte compacto si su suporte:*

$$supp(f) = \overline{\{x \in X | f(x) \neq 0\}}$$

es un subconjunto compacto de X. El conjunto de funciones continuas de soporte compacto de X se denota como:

$$\mathcal{C}_c(X) = \left\{ f : X \to \mathbb{R} \Big|_{supp(f) \text{ es un subconjunto compacto de } X}^{f \text{ es continua y}} \right\}$$

Entonces una función continua $f : X \to \mathbb{R}$ pertenece a $\mathcal{C}_c(X)$ si y solo si existe un subconjunto compacto $K \subset X$ de tal forma que $f(x) = 0$ para todo $x \in X \setminus K$. También implica que $\mathcal{C}_c(X)$ es un espacio vectorial real (Salamon, 2020).

Definición 12.4.8. ***Funcional Real Positivo****: se dice que un funcional lineal $L : \mathcal{C}_c(X) \to \mathbb{R}$ es positivo si:*

$$f \geq 0 \Rightarrow L(f) \geq 0$$

para todo $f \in \mathcal{C}_c(X)$ (Salamon, 2020).

Teorema 12.4.9. *Teorema de Representación de Riesz*

Versión 1 *(Salamon, 2020)*

Sea $L : \mathcal{C}_c(X) \to \mathbb{R}$ un funcional lineal positivo. Entonces se cumple lo siguiente:

- *Existe una única medida de Radon $\mu_0 : \mathcal{B} \to [0, \infty]$ de tal forma que $L_{\mu_0} = L$.*
- *Existe una única medida de regular externa de Borel $\mu_1 : \mathcal{B} \to [0, \infty]$ de tal forma que $L_{\mu_1} = L$.*
- *Las medidas de Borel μ_0 y μ_1 definidas con anterioridad coinciden en todos los conjuntos compactos y todos los conjuntos abiertos. Además $\mu_0(B) \leq \mu_1(B)$ para todo $B \in \mathcal{B}$.*
- *Sea $\mu : \mathcal{B} \to [0, \infty]$ una medida de Borel que es regular interna sobre conjuntos abiertos. Entonces $L_\mu = L$ si y solo si $\mu_0(B) \leq \mu(B) \leq \mu_1(B)$ para todo $B \in \mathcal{B}$.*

Versión 2 *(Calin, 2020)*

Sea F un funcional lineal acotado sobre $\mathcal{C}(K)$ donde K es un compacto, entonces existe una única medida con signo de Borel μ sobre K que cumple con:

$$F(f) = \int_K f(x) d\mu(x), \qquad \forall f \in \mathcal{C}(K)$$

Si se reemplaza la acotación por la positividad del funcional entonces tenemos el siguiente resultado en donde la medida con signo se convierte en una medida de Borel.

Versión 3 *(Calin, 2020)*

Sea F un funcional lineal positivo sobre $\mathcal{C}(K)$ donde K es un compacto, entonces existe una única medida de Borel μ sobre K que cumple con:

$$F(f) = \int_K f(x) d\mu(x), \qquad \forall f \in \mathcal{C}(K)$$

12.5. Análisis Funcional

Teorema 12.5.1. *Si un funcional lineal $f(x)$ es continuo en un punto $x_0 \in X$ donde X es un espacio, entonces es continuo en todo X (Kolmogorov y Formin, 1954).*

Teorema 12.5.2. *Para las funcionales lineales las condiciones de continuidad y acotación, son equivalentes. Un funcional lineal es acotado si y solo si es continuo (Kolmogorov y Formin, 1954).*

Teorema 12.5.3. *Teoremas de Hahn - Banach*

Versión 1*:*

Suponga que M es un subespacio real de un espacio vectorial real X, $p : X \to \mathbb{R}$ satisface lo siguiente:

$$p(x + y) \leq p(x) + p(y)$$

y

$$p(tx) = tp(x)$$

si $x, y \in X, t \geq 0$, también se tiene una función lineal $f : M \to \mathbb{R}$ de tal forma que $f(x) \leq p(x)$ sobre M entonces:

Existe un funcional lineal $L : X \to \mathbb{R}$ que cumple:

$$L(x) = f(x)$$

$x \in M$ y también se encuentra acotado por:

$$-p(x) \leq L(x) \leq p(x)$$

cuando $x \in X$ (Kolmogorov y Formin, 1954), (Rudin, 1991).

Versión 2:

Suponga que M es un subespacio de un espacio vectorial X, p es una seminorma sobre X y que f es un funcional lineal sobre M de tal forma que:

$$|f(x)| \leq p(x) \qquad x \in M$$

entonces f se extiende a un funcional lineal L sobre X que satisface lo siguiente:

$$|L(x)| \leq p(x) \qquad x \in X$$

(Kolmogorov y Formin, 1954), (Rudin, 1991)