



**UNIVERSIDAD DE CONCEPCIÓN
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA**



Sistema de seguimiento BLE de múltiples tags para uso de interiores

POR

Jocsan Magdiel Vergara Blanco

Memoria de Título presentada a la Facultad de Ingeniería de la Universidad de Concepción para optar al título profesional de Ingeniero Civil Electrónico

Profesor Guía
Miguel Ernesto Figueroa Toro

Profesor(es) Co-Guía
Sergio Kendrick Sobarzo Guzmán
Mario Rubén Medina Carrasco

Profesional Supervisor
Pablo Alfredo Verdugo Rubilar

Octubre 2022
Concepción (Chile)

©2022 Jocsan Magdiel Vergara Blanco

©2022 Jocsan Magdiel Vergara Blanco

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento.

Escriba aquí su dedicatoria. Esta página es opcional, y no debiese exceder de una página.

Si está presente, corresponde a la página iii del documento.

Agradecimientos

Escriba aquí sus agradecimientos. Esta página es opcional, y no debiese exceder de una página. Si está presente, corresponde a la página iii o iv del documento, dependiendo de si su documento tiene una página de dedicatoria.

Sumario

Esta página contiene el resumen de su memoria en español. Su extensión no debe superar una página.

Summary

This page contains your thesis's summary, in English. Its length must not exceed one page.

Contenido

Capítulo 1. Introducción.....	1
1.1. Introducción General	1
1.2. Trabajos previos.....	2
1.2.1. RSSI.....	2
1.2.2. AoA	3
1.2.3. Redes Neuronales	4
1.3. Discusión.....	4
1.4. Hipótesis de trabajo.....	5
1.5. Objetivos	5
1.5.1. Objetivo general	5
1.5.2. Objetivo específico.....	5
1.5.3. Alcances y limitaciones.....	6
Capítulo 2. Marco teórico.....	7
2.1. Introducción	7
2.1.1. GNSS.....	7
2.1.2. BLE	7
2.2. Técnicas de Posicionamiento	9
2.2.1. RSSI	9
2.2.2. Trilateración	10
2.2.3. Triangulación	11
2.3. Redes neuronales	12
2.3.1. Introducción	12
2.3.2. Paradigma del aprendizaje	12
2.3.3. Estructura de la red.....	12
2.3.4. Función de coste.....	16
2.3.5. Descenso del gradiente.....	16

2.3.6. Entrenamiento	17
Capítulo 3. Avances	19
3.1. Redes neuronales	19
3.1.1. Dataset.....	19
3.1.2. Redes neuronales.....	20
Referencias	25

Lista de tablas

Tabla 1. Estructura del archivo CSV parte 1	19
Tabla 2. Estructura del archivo CSV parte 2	19

Lista de figuras

Figura 2.1. Representación de sistemas AoA y AoD [2]	8
Figura 2.2. Capa de comunicación Bluetooth con la extensión de tono constante [2].....	9
Figura 2.3. Representación de un arreglo de antenas y la señal recibida [3]	9
Figura 2.4. Ilustración de la trilateración	10
Figura 2.5. Ilustración de la trilateración	11
Figura 2.6. Función Escalón	13
Figura 2.7. Función Sigmoid	13
Figura 2.8. Función tangente hiperbólico	14
Figura 2.9. Función ReLU	14
Figura 2.10. Representación gráfica de una neurona.	15
Figura 2.11. Representación de una red neuronal	15
Figura 3.12. Error de la red neuronal lineal	21
Figura 3.13. Gráfico de predicciones vs los valores reales obtenidos por la red neuronal lineal	21
Figura 3.14. Error en el entrenamiento de la red neuronal no lineal.....	22
Figura 3.15. Gráfico de predicciones vs los valores reales obtenidos por la red neuronal no lineal	22
Figura 3.16. Error en el entrenamiento de la red neuronal convolucional	23
Figura 3.17. Gráfico de predicciones vs los valores reales obtenidos por la red neuronal convolucional..	23
Figura 3.18. Gráfico de la precisión del modelo convolucional	24

Capítulo 1. Introducción

1.1. Introducción General

Conocer la ubicación exacta de una persona, mascotas, objetos en determinados escenarios es importante, nos permite tener tranquilidad, seguridad frente a determinadas situaciones, además de garantizarnos cierto control de las cosas. Siendo potenciado aún más con las nuevas tecnologías y el aumento en el desarrollo de tecnologías IoT o internet de las cosas, donde se busca obtener esta información de forma sencilla y rápida, pero sin dejar de lado una característica importante, la precisión.

Desde hace años se ha perfeccionado el uso de satélites, que orbitan por toda la tierra para poder ubicarnos en espacios abiertos, estos poseen un error de unos pocos metros o inclusive de centímetros, no obstante, es necesario tener un mínimo de tres satélites sincronizados entre ellos para determinar la geo posición de un objeto, por medio de sistema de posicionamiento global (Global Positioning System GPS).

Esta técnica de localización no funciona muy bien en espacios cerrados, decayendo su precisión e incluso siendo en algunos casos incapaz de determinar la ubicación correcta. Actualmente se implementan varias soluciones para obtener la posición en espacios cerrados como lo son identificador por radiofrecuencia (Radio Frequency Identification, RFID), bandas ultra anchas (Ultra Wireband, UWB), redes de área local inalámbricas (Wireless Local Area Network, WLAN) y Bluetooth de bajo consumo energético (Bluetooth Low Energy, BLE) por nombrar algunos ejemplos.

Se usará un hardware de la marca U-blox con la tecnología BLE 5.1, este posee un bajo consumo energético y el coste de esta tecnología se ha vuelto más accesible. Su funcionamiento se basa en radiogoniometría Bluetooth, el cual envía la información desde una baliza a un arreglo o conjunto de antenas y obtiene el ángulo. Una característica importante implementada en la revisión 5.1 es la llamada, ángulo de llegada (Angle of Arrival, AoA), esta consiste en determinar el ángulo de incidencia al arreglo de antena. Esta característica nos abre las puertas a un conjunto de posibilidades, no obstante, también se presentan dificultades, si un objeto se interpone entre la baliza y las antenas, se verá alterada la medición y dificultándose el poder determinar la ubicación correcta.

1.2. Trabajos previos

Se realiza una recopilación de trabajos previos, enfocada a conocer aspectos teóricos y fundamentales sobre el tema desarrollado.

1.2.1. RSSI

- M. Ji, J. Kim, J. Jeon and Y. Cho, "Analysis of positioning accuracy corresponding to the number of BLE beacons in indoor positioning system," 2015 17th International Conference on Advanced Communication Technology (ICACT), 2015, pp. 92-95, doi: 10.1109/ICACT.2015.7224764, [6].

En este trabajo se presenta la siguiente metodología de trabajo con señales BLE y Wi-Fi para medir RSSI, de estas señales a determinadas distancias, a 1m metro de distancia hasta llegar a los 13 metros, en cada punto de referencia se tomaron los valores durante un minuto y después al cambiar de punto se esperan dos minutos para recabar los datos, también se menciona una línea de visión despejada, se utilizaron 4 Beacons para las pruebas y dos módulos Wifi-AP. Obteniendo el error en metros y también realizan simulaciones para ver si la estimación del error justificaría un aumento de Beacons llegando a la conclusión de que existen un punto donde el error disminuye de forma no tan significativa, pudiendo ser el caso de que no justifique la infraestructura a implementar.

- A. Noertjahyana, I. A. Wijayanto and J. Andjarwirawan, "Development of Mobile Indoor Positioning System Application Using Android and Bluetooth Low Energy with Trilateration Method," 2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIIT), 2017, pp. 185-189, doi: 10.1109/ICSIIT.2017.64, [7].

En este trabajo tenemos el uso de la trilateración para el cálculo de la distancia y posición de objetivo, cabe resaltar que se utilizar un Smartphone con sistema operativo Android, el cual cuenta con GPS, en su conjunto la aplicación que ellos implementan es que por medio de la señal Bluetooth infieren la posición en el interior de un edificio y con ayuda del GPS determinan las coordenadas, esta información es subida a un servidor a una base de Datos MySQL para su posterior recuperación, dentro de las mediciones obtienen los valores de RSSI sin obstáculos y con una puerta de madera.

- Y. Shi, Y. Long, F. Lu, Z. Xu, X. Xiao and S. Shi, "Indoor RSSI Trilateral Algorithm Considering Piecewise and Space-Scene," 2017 IEEE International Conference on Smart

Cloud (SmartCloud), 2017, pp. 278-282, doi: 10.1109/SmartCloud.2017.52., [11].

En este trabajo se ve un primer paso en la optimización de los parámetros que determinan el valor de distancia en función del RSSI, las condiciones donde se realizaron la obtención de los datos es un espacio de ancho 3,5 m y una longitud de 30m, en este lugar se realizaron las medición en función de 50 cm distancia sucesivamente , una vez se obtienen los valores se busca obtener los coeficientes para la ecuación 2.1, siendo optimizado con las herramientas de Matlab para poder interpolar los datos de la medición y obtener finalmente una función por parte, se realizan pruebas en un espacio de 3.5m por 9m obteniendo una error de entre 0.9-1.5m, siendo una mejora que en el caso de solo usar la ecuación original, como en los documentos anteriormente citados se realiza trilateración para medir la distancia.

1.2.2. AoA

- P. Andersson and L. Persson, "Evaluation of Bluetooth 5.1 as an Indoor Positioning System," Dissertation, 2020, [8].

En este trabajo se realiza la medición de distancia por medio de la medida de los ángulos y se utiliza la triangulación, para ellos se utiliza el LAUCNHXL-CC2640R2F que es un kit de desarrollado, dos antenas BOOSTXL-AOA y un módulo IMU ADXL343, fabricado por Texas Instruments Inc. Estas hacen uso de la tecnología Bluetooth 5.1, y la comunicación utilizada es UART. Se realizan las pruebas sin y con el filtro EKF el cual se suele utilizar en sistemas cuyos sensores presentan ruido de tipo gaussiano, permitiendo corregir por medio de la retroalimentación de este filtro predictivo. El enfoque analizado en este documento es el tiempo de respuesta, eficiencia energética y la precisión en el tiempo.

- P. Sambu and M. Won, "An Experimental Study on Direction Finding of Bluetooth 5.1: Indoor vs Outdoor," 2022 IEEE Wireless Communications and Networking Conference (WCNC), 2022, pp. 1934-1939, doi: 10.1109/WCNC51071.2022.9771930, [9]

En este trabajo se realizaron pruebas con dos TI transmisores Bluetooth CC26X2R LaunchPad y dos antenas BOOSTXL-AOA ambos fabricados por Texas Instruments Inc. se hace uso de los ángulos de llegada para determinar la distancia, además el enfoque de este trabajo es analizar las diferencias con respecto a los ángulos y a la distancia obtenida con la que corresponde, tanto en interiores como en el exterior. Los resultados obtenidos hablan de una mayor precisión en el exterior frente a las mediciones tomadas en el interior un 73% mejores mediciones en el exterior, el error lo

define con la distancia euclidiana. De igual forma se menciona que en el caso de ángulos pequeños cercanos a 0° presentan una mayor desviación estándar que para ángulos mayores a 35° asumiendo que se debe a los problemas de retardo de fase, tomando ángulos mayores a 15° para evitar ese error. En los resultados obtenidos el error angular medio para el entorno exterior es de 0.48° con una desviación estándar de 0.28 y para interiores tenemos un del ángulo medio de 1.83° con la desviación estándar de 0.48° . Igualmente, el error en posición 2D (eje x e y) es de 36,5cm para interiores y 22cm para exteriores, esta prueba y mediciones se obtuvieron al generar una cuadrícula de 5 subdivisiones de 2mx2m cada una.

- P. Sambu and M. Won, "An Experimental Study on Direction Finding of Bluetooth 5.1: Indoor vs Outdoor," 2022 IEEE Wireless Communications and Networking Conference (WCNC), 2022, pp. 1934-1939, doi: 10.1109/WCNC51071.2022.9771930, [9].

1.2.3. Redes Neuronales

- P. Babakhani, T. Merk, M. Mahlig, I. Sarris, D. Kalogiros and P. Karlsson, "Bluetooth Direction Finding using Recurrent Neural Network," 2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2021, pp. 1-7, doi: 10.1109/IPIN51156.2021.9662611, [10]

En este trabajo propone el uso de red neuronal recurrente (RNN) para mejorar la estimación de AoA, se considera este problema como un problema de continuidad temporal, siendo necesario una red neuronal capaz de tomar decisiones en base a los datos obtenidos anteriormente, también se usa una unidad recurrente cerrada debido a las exigencias del problema a la hora de determinar la posición en el tiempo de un objetivo en movimiento y motivados también por el costo computacional. Cabe resaltar el uso de PDDA para el análisis las características del espectro a través de los canales. También es importante señalar que para el entrenamiento de la RNN se usa un software de simulación RayRacing, Aktair WinProp simulando con cuatro antenas a una potencia de +4dBm.

1.3. Discusión

En el caso de [2]. El método de EKF no es muy efectivo en sistemas que posean demasiadas variables, añadir por ejemplo demasiadas antenas o tag, pues la dinámica del problema crece, otro punto en contra es el tener que resolver sistemas no lineales, lo cual nos lleva a tener que linealizar en torno al un punto. El costo computacional es alto a su vez por tener que calcular el Jacobinao en cada paso, además de la matriz inversa necesaria para hacer esta operación, lo cual podría ser

imposible de resolver, hablando en tiempos de ejecución. A estos problemas se añade el no poder eliminar el ruido de las múltiples trayectorias, la existencia de datos deficientes aumenta la desviación estándar. Se aprecia que la precisión obtenida con RSSI es menor en relación con la obtenida con AoA y es más sensible a las perturbaciones.

En el caso de [10] tenemos una puerta a seguir experimentando con redes neuronales, en búsqueda a la solución al problema de la propagación por múltiples trayectorias.

En el transcurso de este documento buscaremos añadir la intensidad de la señal RSSI a la solución del problema, esto debido a que Bluetooth trabaja con frecuencias altas, es por esto que las señales son altamente direccionales, al cambiar su dirección la potencia de la señal descende, siendo un buen indicador de que tenemos problemas con la LOS.

1.4. Hipótesis de trabajo

Como se vio a lo largo de la revisión bibliográfica, usar solamente la intensidad de la señal no es suficiente para determinar la distancia, al implementar las características añadidas en Bluetooth 5.1 se pueden obtener mejores resultados, pero en presencia de objetos que obstaculicen la línea de visión se obtienen resultados incorrectos en un cierto número de grados.

Se plantea el poder implementar redes neuronales con el fin de corregir estas desviaciones, tomando en consideración las mediciones realizadas en tiempos anteriores y determinando si el desplazamiento realizado es correcto, se busca por tanto poder implementar una RNN LSTM debido a su propiedad de tomar decisiones en base a valores pasados, pero también se busca añadir como valor de entrada a la red neuronal el RSSI, el cual podría ser un indicador de cuanto se ve afectada la señal.

1.5. Objetivos

1.5.1. Objetivo general

Diseñar un sistema que utilice múltiples antenas para localizar múltiples tags BLE dentro de una casa.

1.5.2. Objetivo específico

- Elaborar una base de datos.
- Seleccionar y entrenar una red convolucional para detectar errores en la medición.
- Seleccionar y entrenar una red neuronal para corregir las mediciones basado en la posición anterior.
- Evaluar la precisión y desempeño computacional de la implementación.

- Desarrollar una interfaz gráfica para poder visualizar los resultados obtenidos.

1.5.3. Alcances y limitaciones

- Se espera un error máximo en la medición de 50 cm.
- El entrenamiento de las redes se realizará utilizando la plataforma de Google Colab o AWS de ser necesario.
- Se utilizará el lenguaje de programación Python para realizar la implementación.
- Se implementarán los modelos en el ordenador personal del alumno, un Lenovo P43s.
- El lugar de roma de puntos para el dataset y lugar de trabajo serán en las dependencias del alumno, en las oficinas de la empresa SICOM, en el laboratorio de VLSI en las dependencias de la universidad.

Capítulo 2. Marco teórico

2.1. Introducción

2.1.1. GNSS

El sistema global de navegación por satélite (Global Navigation Satellite System, GNSS), consiste en un conjunto de satélites interconectados y sincronizados, capaces de determinar la posición y localización de un objetivo en cualquier parte de la tierra. Para determinar su posición es necesario medir su distancia relativa con respecto a tres satélites, estos deben saber su propia posición todo el tiempo y poder sincronizarse entre sí. En caso de querer determinar la altitud, es necesario añadir un cuarto satélite. Una vez los satélites están sincronizados por medio de la trilateración obtenemos la menor distancia, esta consiste en extender una circunferencia en cuyo centro se encuentra el satélite, hasta poder interceptar las tres circunferencias generadas, por medio del radio obtenido se determina la menor distancia, a su vez, la localización es inferida al determinar su posición con respecto a los satélites, pues como se mencionó anteriormente los satélites conocen su localización todo el tiempo.

Actualmente contamos con diferentes GNSS en funcionamiento

1. El sistema de posicionamiento global (Global Positioning System, GPS) el cual corresponde al sistema de satélites creados por los Estados Unidos.
2. El sistema global de navegación por satélite (Global Navigation Satellite System, GLONASS), corresponde al sistema de navegación satelital pertenecientes a Rusia.
3. El sistema de navegación por satélites BeiDou (The BeiDou Navigation Satellite System, BDS), es el sistema de navegación desarrollado por China.
4. Galileo, es el sistema creado por la Unión Europea.

2.1.2. BLE

La tecnología BLE consiste en redes inalámbricas interconectadas, las cuales operan dentro del rango de frecuencia de banda de radio industriales, científicas y médicas (ISM) de 2,4 Ghz para telecomunicaciones en general. Hasta antes de las integraciones realizadas en la especificación de Bluetooth 5.1, los sistemas de posicionamiento en interiores (indoor positioning system, IPS) estaban limitados al uso de la intensidad de la señal recibida (received signal strength indicator, RSSI) para estimar la distancia [1]. En esta revisión fueron añadidas dos funciones de radiogoniometría, las cuales consisten en un sistema electrónico capaz de determinar la dirección de donde viene una señal de

radio. Una de ellas determina el ángulo de salida (Angle of Departure, AoD) y la otra determina el ángulo de llegada de la señal (Angle of Arrival, AoA).

Para un sistema que implementa AoD es necesario que el transmisor posea un arreglo de antenas para determina la dirección del receptor hacia donde llegue la señal. Para el sistema que implementa AoA para determinar la dirección de la señal, es necesario que el receptor posea el arreglo de antenas con el cual es capaz de determinar el ángulo de incidencia de la señal de radiofrecuencia.

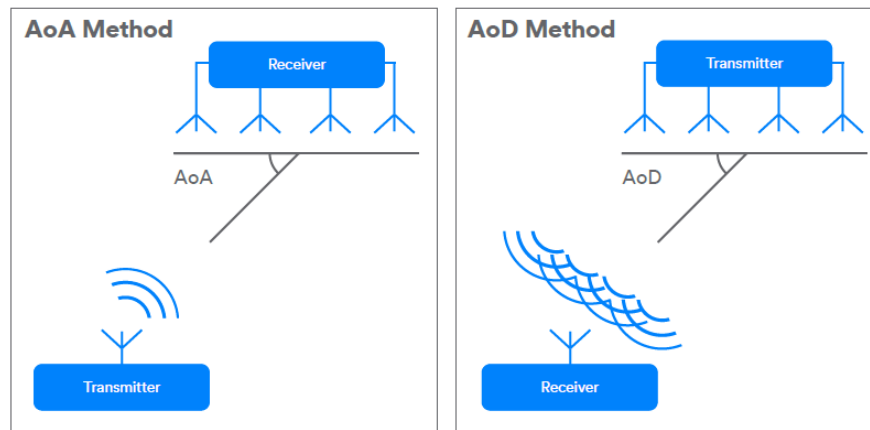


Figura 2.1. Representación de sistemas AoA y AoD [2]

Para poder obtener el ángulo de incidencia es necesario realizar un muestreo en fase y cuadratura (IQ). Se busca poder expresar una señal en término de dos formas de onda a esto se le denomina función de cuadratura. A la frecuencia de muestreo de una señal analógica se le descompone en dos señales, luego se aplican el teorema de Nyquist a la señal de entrada con las frecuencias de muestreo generadas, obteniendo en este caso dos señales diferentes.

Una vez la señal de radio frecuencia incide sobre el arreglo de antenas, el dispositivo receptor ve una diferencia de fase de la señal debido a la diferencia de distancia entre cada una de las antenas del conjunto. El dispositivo receptor toma muestras IQ de la señal mientras cambia entre las antenas activas en el arreglo. Según los datos de la muestra de IQ, el aparato receptor puede calcular la dirección relativa de la señal.

Las señales enviadas por radiofrecuencia Bluetooth deben tener una estructura definida, esta hace parte del protocolo de comunicación y permite establecer los enlaces de comunicación, además del posterior tratamiento de la señal, donde se obtienen datos de identificación entre los participantes, datos propios de la comunicación y información necesaria para la validación y corrección de los datos, como lo es la verificación de redundancia cíclica (cyclic redundancy check, CRC), sin embargo, para poder realizar IQ es necesario añadir más información al protocolo de comunicación, siendo esta la

extensión de tono constante (constant tone extension, CTE), que es una señal de frecuencia constante que se añade al final del protocolo, no lleva información modulada sino que, informa del canal usado para enviar la señal correspondiente, siendo inferido por la frecuencia que posee.

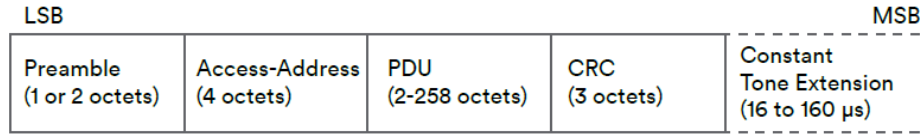


Figura 2.2. Capa de comunicación Bluetooth con la extensión de tono constante [2]

Desde el protocolo Bluetooth 5.0, las bandas destinadas para publicidad son los canales 37, 38 y 39, estos son utilizados para enviar los datos tanto en AoA como en AoD, se escogieron esos canales a fin de disminuir las posibilidades de colisiones de paquetes enviados por otros dispositivos que operen dentro de la banda ISM, también son utilizados en orden ascendente y tienen un retardo de envío aleatorio entre 0 y 10 ms buscando evitar posibles colisiones entre paquetes enviados.

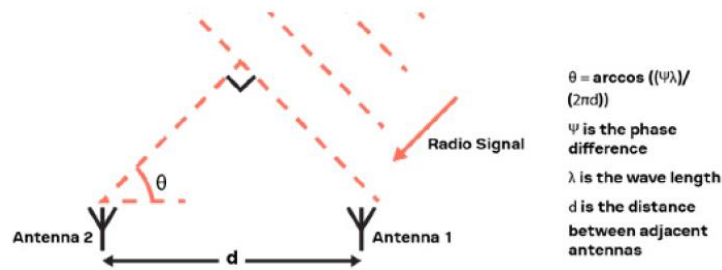


Figura 2.3. Representación de un arreglo de antenas y la señal recibida [3]

2.2. Técnicas de Posicionamiento

2.2.1. RSSI

Este método consiste en medir la intensidad de la señal recibida por el receptor y aproximar la distancia entre el transmisor y el receptor con esta información. Esta técnica funciona muy bien cuando no existe interferencia y/o objetos en la línea de visión, de igual manera el transmisor debe permanecer quieto junto con el receptor. A su vez, existen pérdidas en la intensidad de la señal debido a la propagación de esta en el medio, además de las múltiples trayectorias que puede tomar la señal. Por medio de la siguiente ecuación podemos obtener el valor de RSSI en función de la distancia.

$$RSSI = -(10n \log_{10} d + A)$$

Siendo el RSSI la medida de la intensidad de la señal, n es el exponente de pérdida de trayectorias el cual depende del entorno, siendo necesario calcularlo para cada configuración, d

corresponde a la distancia entre el transmisor y el receptor. A corresponde a la potencia absoluta medida en dBm, se le asigna el valor del RSSI a 1 metro de distancia.

En [5] se menciona que, en espacios libre, el color de n entre antenas isotrópicas se comporta de forma proporcional al cuadrado de la distancia según la siguiente ecuación.

$$n = \left(\frac{4\pi d}{\lambda} \right)^2$$

También se menciona que la reflexión del suelo, el cual modifica el valor de n entre 2 y 4, siendo el caso ideal el valor de 2 para un espacio libre.

2.2.2. Trilateración

Se utiliza para determinar las posiciones relativas de los objetivos, es el método utilizado en los sistemas de GNSS y también en los métodos de IPS usando el RSSI. Se determina la posición de un objetivo, generando circunferencias en cuyo centro se encuentra cada antena o satélite según sea el caso, estas deben interceptarse, en este momento el radio deja de crecer y se obtiene la distancia por medio del radio de la circunferencia entre el transmisor y el receptor. Al saber la posición de los satélites o las antenas se puede determinar la posición del objetivo.

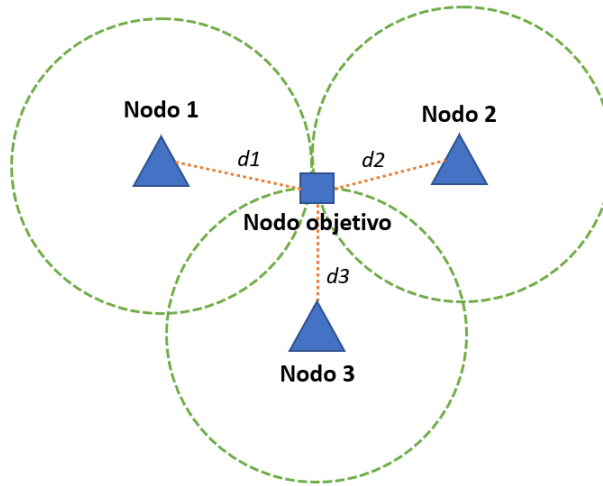


Figura 2.4. Ilustración de la trilateración

Esto se expresa con un sistema de ecuaciones:

$$\begin{bmatrix} (x_1 - x_e)^2 + (y_1 - y_e)^2 \\ (x_2 - x_e)^2 + (y_2 - y_e)^2 \\ (x_3 - x_e)^2 + (y_3 - y_e)^2 \end{bmatrix} - \begin{bmatrix} d_1^2 \\ d_2^2 \\ d_3^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Nuestro objetivo es obtener el valor de x_e y y_e siendo necesario resolver el sistema de ecuaciones. No obstante, es posible que los puntos obtenidos no se intercepten. En [2] proponen

buscar el resultado que genere el error más pequeño, convirtiendo el problema en uno de optimización.

El problema mencionado anteriormente se puede generar al tener errores en las mediciones o una precisión deficiente, un debilitamiento de la intensidad de la señal o cambios en el medio, alguien pasa por la línea de visión o la existencia de reflexiones por parte de la señal, también, pueden existir retardos en la medición y posterior almacenamiento, algún problema en el hardware, propios de la fabricación del dispositivo, entre otras.

2.2.3. Triangulación

Es el método de posicionamiento que hace uso de la ley de seno. Es necesario colocar dos nodos cuya distancia entre ellos sea conocida. Luego se mide la dirección o ángulo desde donde viene la señal, con estos datos podemos obtener la distancia entre cada una de los nodos y el transmisor de la señal.

$$\frac{A}{\sin(\alpha)} = \frac{B}{\sin(\beta)} = \frac{C}{\sin(\gamma)}$$

Considerando la figura 5

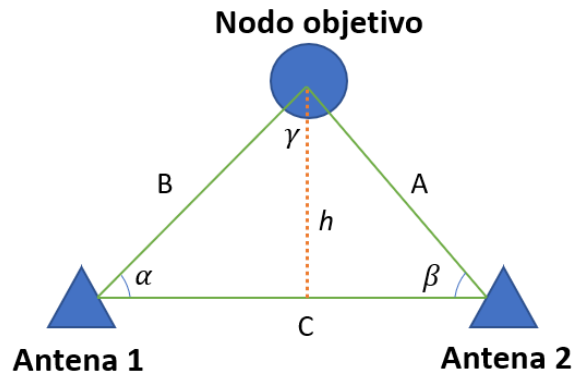


Figura 2.5. Ilustración de la trilateración

Tenemos que la distancia A y B se pueden obtener por medio de las siguientes ecuaciones

$$A = \frac{C \sin(\alpha)}{\sin(\gamma)}$$

$$B = \frac{C \sin(\beta)}{\sin(\gamma)}$$

Y la altura del triángulo se obtiene con la siguiente ecuación

$$h = \frac{C \sin(\beta) \sin(\alpha)}{\sin(\beta + \alpha)}$$

2.3. Redes neuronales

2.3.1. Introducción

El aprender algo nuevo, es una característica que muchos de nosotros realizamos de manera inconsciente, donde una infinidad de procesos biológicos y físicos se ven involucrados para concretar con la tarea de aprender, siendo además esta característica la cual nos permite adquirir nuevas habilidades y destrezas. Una vez entendemos y asimilamos el contenido podemos resolver problemas futuros en base al conocimiento adquirido, ante esta idea es que surge el aprendizaje automático (Machine Learning, ML) el cual pertenece a una rama de la inteligencia artificial, en donde se busca poder construir programas de computadoras capaces de mejorar automáticamente adquiriendo experiencia.

La lógica a la hora de realizar un programa de computador es entregarle los patrones o reglas específicas al programa y que, en base a los datos de entrada, se obtengan resultados a la salida. No obstante, cuando la cantidad de datos es demasiado grande, obtener dichos patrones o reglas se torna complejo, más aún si los datos poseen una componente no lineal o no poseen un patrón tan definido, otro factor importante es el tiempo del cual dispone la persona para encontrar dichos patrones, ante esto se hace aún más necesario poder construir programas capaces de extraer estos patrones y tendencias de los datos.

2.3.2. Paradigma del aprendizaje

Para que una red neuronal aprenda es necesario realizar un entrenamiento, el cual consiste en entregarle una serie de datos a la red con el fin de que ella encuentre los patrones inherentes en estos. Dependiendo de los datos entregados se pueden separar en dos grupos mayoritariamente, datos con etiquetas (labels) y sin estas, las etiquetas contienen la respuesta correcta a la solución del problema. Por tanto, el aprendizaje supervisado es en cual los datos poseen las etiquetas, mientras que el aprendizaje no supervisado no las posee.

2.3.3. Estructura de la red

La unidad básica en la construcción de una red neuronal se llama neurona, esta consiste en la suma ponderada de los valores a la entrada de la neurona y una función de activación, para realizar la suma ponderada, a cada entrada se le asignar un valor numérico denominado peso (weight, w) el cual indica el grado de dependencia de la salida con determinada entrada, dentro de esta suma ponderada se añade una entrada que es independiente denominada sesgo (bias, b). Se le suele asignar el valor

negativo de un umbral.

La función de activación añade la no linealidad a la suma ponderada, deformando el hiperplano generado por F, existen varias funciones de activación y dependiendo de la aplicación es que se escoge entre una de ellas, por ejemplo, tenemos la función escalón, sigmoide, Tangente hiperbólico y la función ReLU.

La función escalón nos permite llevar los valores de entrada negativos a 0 y los valores positivos quedan iguales a 1.

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

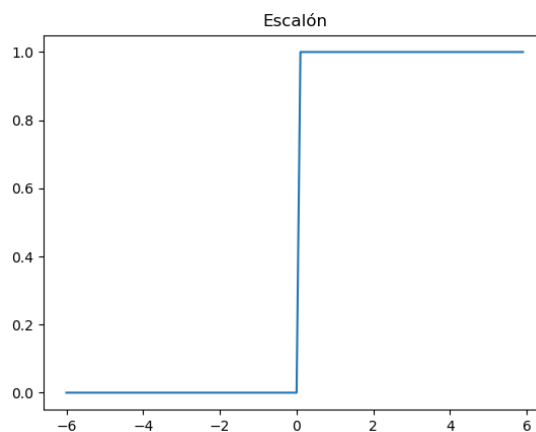


Figura 2.6. Función Escalón

La función sigmoide nos permite transformar los valores de entrada desde -inf hasta inf a valores dentro del rango de 0 hasta 1.

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

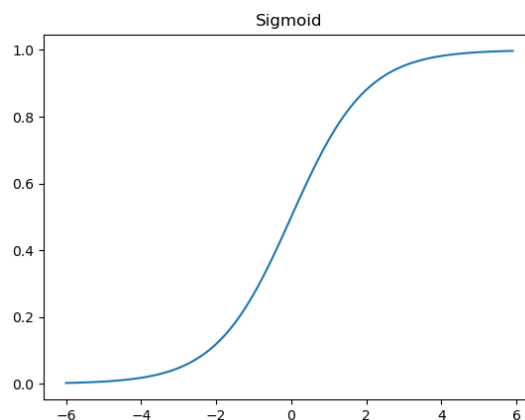


Figura 2.7. Función Sigmoid

La función TANH o tangente hiperbólico posee un comportamiento similar a la función

sigmoide, pero a su salida tenemos valores entre -1 a 1.

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

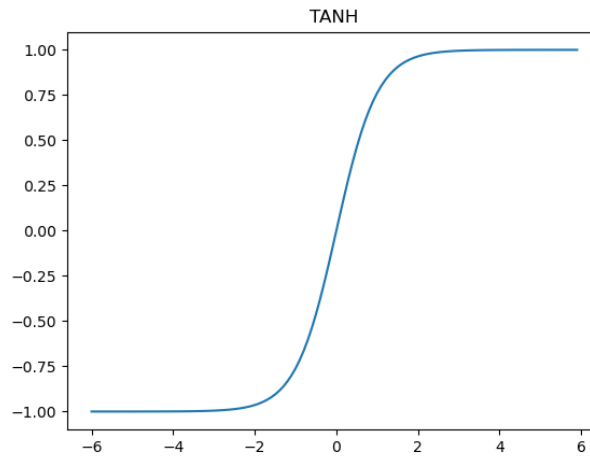


Figura 2.8. Función tangente hiperbólico

La función de activación ReLU (Rectified Linear Unit) es una transformación no lineal que activa la neurona solo si los valores a la entrada (input) están sobre 0, está nos permite descartar los valores negativos de un conjunto de datos de ser necesario.

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

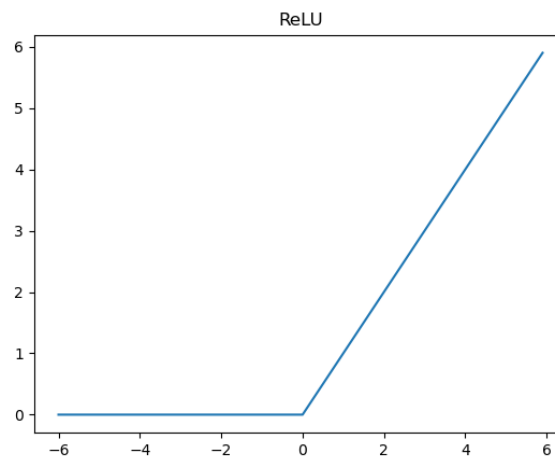


Figura 2.9. Función ReLU

La función de activación Softmax o función exponencial normalizada, se suele usar en la capa final en las redes neuronales basadas en clasificadores, con esta función obtenemos la distribución de probabilidad de cada una de las clases soportadas en el modelo, esto se logra al comprimir un vector K-dimensional, z, tal que los valores a la salida están en el rango entre 0 y 1.

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

De esta manera, tenemos finalmente la unidad básica de una red neuronal, la neurona se define matemáticamente como.

$$F = f(w_1X_1 + W_2X_2 + b)$$

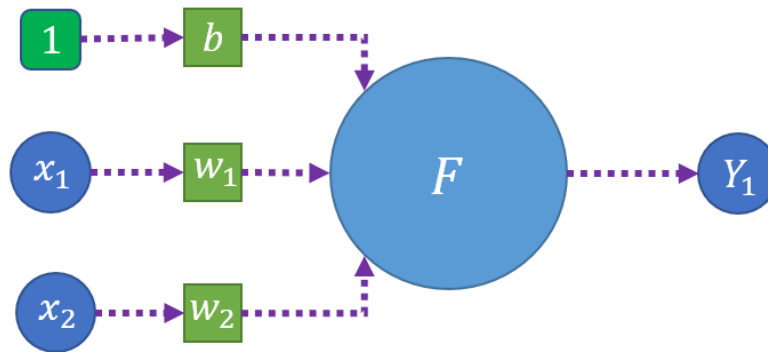


Figura 2.10. Representación gráfica de una neurona.

Para resolver problemas más complejos y generar un aprendizaje profundo es necesario aumentar el número de neuronas e interconectarlas entre sí, a esta interconexión de neuronas se les conoce como redes neuronales.

Las redes neuronales se construyen por capas, una capa está compuesta por neuronas todas en un mismo nivel, también cuando tenemos una red neuronal se suele agrupar las capas de la siguiente forma, la capa de entrada (Input layer) es donde se reciben los datos de entrada, tenemos las capas ocultas (Hidden layer) en donde se realizan las operaciones matemáticas y la capa de salida (Output layers) donde se obtienen como dato de salida las predicciones de la red.

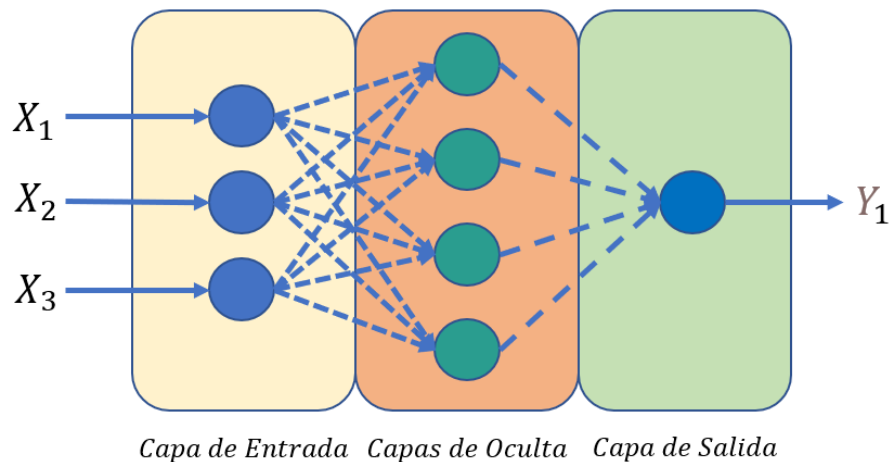


Figura 2.11. Representación de una red neuronal

2.3.4. Función de coste

La función de coste o también llamada función de pérdida (loss function o cost function), se encarga de medir la distancia entre el valor real y el valor predicho por la red, permitiéndonos evaluar el error generado al realizarse una predicción, se suele obtener un valor positivo de esta función además mientras más pequeño es el valor obtenido, mejores son las predicciones de la red.

Dependiendo del tipo de problema ya sea regresión o clasificación se utiliza una función de coste o otra. Para el caso de regresiones, se utilizan como función de costo el error cuadrático medio o el error absoluto medio, mientras que para el caso necesitar una clasificación se suele emplear log loss o la entropía cruzada (cross entropy).

El error cuadrático medio (mean squared error, MSE) se obtiene realizando la suma de la diferencia al cuadrado entre el valor predicho y el valor real.

$$L(w, b) = \frac{1}{n} \sum_{i=1}^n (y_{pred}(i) - y_{real}(i))^2$$

El error medio absoluto (mean absolute error, MAE) consiste en promediar el error absoluto, se obtiene realizando el valor absoluto de la diferencia entre el valor predicho y el valor real.

$$L(w, b) = \frac{1}{n} \sum_{i=1}^n |y_{pred}(i) - y_{real}(i)|$$

Cuando se entrena un modelo utilizando el MSE como función de coste, este aprende a predecir la media de los valores de salida, en el caso de hacerlo con MAE aprenderá a predecir la mediana de los valores a la salida.

Log loss o cross entropy loss, para problemas de clasificación, la capa de salida utiliza es la función de activación Softmax, de esta forma obtenemos la probabilidad de que determinada salida sea la correcta.

$$L_{log}(Y, P) = -\log \Pr(Y|P) = -\frac{1}{N} \sum_{i=0}^{N-1} \sum_{k=0}^{K-1} y_{i,k} \log p_{i,k}$$

2.3.5. Descenso del gradiente

El descenso del gradiente (Gradient descent) es un algoritmo de optimización, el cual nos permite obtener los mínimos globales y locales. Por medio del gradiente de una función podemos determinar la dirección de ascenso en un determinado punto dentro del plano. Al usarlo con el signo negativo podemos determinar la dirección donde se generará el descenso. En la ecuación tenemos que

x_n es el punto donde comenzamos, α corresponde a cuanto nos movemos en una determinada dirección del gradiente negativo, siendo x_{n+1} el punto al cual llegamos.

$$x_{n+1} = x_n - \alpha \nabla f(x_0)$$

En ML, α corresponde al termino determinado como tamaño de paso, si este es pequeño la red aprenderá, pero le tomará mucho tiempo, mientras que, si es demasiado grande, este podría no converger a la solución, pues su valor pasará por sobre este. El uso del descenso del gradiente nos permite ir actualizando los pesos y sesgos del modelo para reducir su error.

No obstante, calcular el error del modelo para todas las iteraciones del entrenamiento es demasiado costoso, por lo cual un método alternativo es el llamado gradiente estocástico (stochastic gradient descent, SGD), este método consiste en dividir los conjuntos de entrenamiento en lotes (batch) y actualizar los parámetros de la red con cada uno. Actualizando los parámetros progresivamente. Una ronda completa de iteraciones sobre todo los batch se llama época. Siendo el número de épocas las veces que la red va a entrenar con el conjunto de datos de entrenamiento.

La retro propagación o propagación hacia atrás de errores (Backpropagation) es un algoritmo implementado en el proceso de entrenamiento, el cual consiste en que, una vez se detecta un error entre el valor esperado y el valor generado por la red neuronal, se busca determinar la causa de ese error, no obstante, en lugar de generar pruebas desde la entrada y comprobar todos los caminos posibles determinando cual neurona es la responsable, se comienza la búsqueda desde la salida en dirección hacia las entradas y se determina que tanto influye determinada neurona en el resultado obtenido, una vez se encuentra el camino que influyo en determinado error se implementa la corrección correspondiente, de esta manera estamos aplicando el descenso del gradiente desde la salida hacia la entrada.

2.3.6. Entrenamiento

Un punto importante para el funcionamiento de una red neuronal es su entrenamiento, lo primero que debemos considerar son los datos, la cantidad de datos es importante y depende de cada problema el número óptimo para obtener buenos resultados, a su vez, estos deben ser separados al menos en una proporción de un 80% para entrenamiento y un 20% para test, a esta técnica se le conoce como (Hold Out). Los datos deben estar distribuidos dentro toda la muestra, datos presentes en el comienzo, al medio y al final con el fin de tener suficientes ejemplos para revisar el progreso de la red, además, no deben poseer demasiada similitud entre sí. De no tomar estas consideraciones es posible terminar con uno de los siguientes problemas.

Una red con valores sub ajustados (Underfitting) es incapaz de obtener buenos resultados y sus

predicciones son incorrectas, debido tener a una pequeña cantidad de datos o un número pequeño de pasos de entrenamiento. Se dice que una red esta sobre ajustada (Overfitting) cuando generaliza las señales de ruido o los datos aislados, esto puede parecer una señal de que la red aprendió de manera correcta, pero a la hora de darle datos que desconoce es incapaz de generalizar la información, puesto que, la red se especializo solo en aprender los datos del entrenamiento, también ocurre cuando los datos con los que contamos poseen demasiada dependencia entre ellos, los datos de entrenamiento y los datos de prueba.

Una técnica que se suele implementar para evitar este problema anteriormente mencionados es el denominado Dropout, el cual consiste en desconectar cierto número de neuronas entre capas.

Capítulo 3. Avances

3.1. Redes neuronales

3.1.1. Dataset

Como se planteo en los objetivos, para poder usar redes neuronales es necesario tener datos para realizar un entrenamiento y posteriormente las prueba sobre las predicciones realizadas por la red. Se realizo un conjunto de datos (dataset) tomando 144 muestras de 2 minutos cada una, se colocó la antena sobre un trípode a una altura de 134 cm y se vario tanto la altura como la distancia del tag.

Se utilizaron dos alturas para el tag, 130 cm y 100 cm, también se vario la distancia entre la antena y el tag desde los 20 cm hasta los 200 cm. Cada punto de medición está a una distancia de 10 cm entre sí. De las 144 muestras, 36 muestras son con LOS libre de objetos y las otras 36 muestras con un objeto entre la antena y el tag. Este objeto obstruye por completo la línea de visión. También es importante mencionar que en cada punto de prueba se repitió la medición dos veces, con el fin de poder generar una dataset más robusto.

Las mediciones fueron realizadas en el domicilio del alumno, almacenadas en archivos con extensión csv para su posterior tratamiento.

La estructura de los archivos csv se puede ver en las siguientes tablas

Tabla 1. Estructura del archivo CSV parte 1

	Hora	Iden_tag	RSSI	Ang_azimuth	Ang_elevacion	Canal
0	2022/09/17/ 16:09:20	:CCF957966B2C	-60	-9	-12	37
1	2022/09/17/ 16:09:20	:CCF957966B2C	-60	-8	-6	37
2	2022/09/17/ 16:09:21	:CCF957966B2C	-56	-9	-12	38
..

Tabla 2. Estructura del archivo CSV parte 2

LOS(si=1,no=0)	Altura_ant(cm)	Distancia_entre_ant_tag(cm)	Altura_tag(cm)	Error_dato_medido
1.0	134	30	130	0.0
1.0	134	30	130	0.0
1.0	134	30	130	0.0
....

No se utilizan todos los datos para el entrenamiento de momento, el caso de la hora será una

variable importante en la red neuronal LSTM puesto que, podemos entregarle los datos históricos a la red para que aprenda.

También, el identificador del tag (Iden_tag) será importante para poder visualizarlo en un futuro, esta de momento debería ser la estructura para ser almacenados en una base de datos.

Se puede observar que además de los datos entregados por la antena que sería el RSSI, el ángulo de Azimuth, el ángulo de elevación y el canal, se añade información de la altura de la antena, su distancia euclidiana entre la antena y el tag y también la altura del tag. También se añade la denominada etiqueta de cada muestra, que corresponden a las respuestas correctas y respuesta que esperamos la red nos entregue, en este caso esperamos sea capaz de determinar si existe algo que se cruce entre la antena y el tag, esta información se añade en última columna en el archivo csv siendo 0 para una línea de visión libre y 1 para una bloqueada.

3.1.2. Redes neuronales

Se implementaron tres tipos de redes, una red lineal sin funciones de activaciones en ninguna de sus capas, una que implementa varias funciones de activación en sus diferentes capas y una red convolucional, siendo esta última la cual entrega mejores resultados.

Antes de poder realizar el entrenamiento de cualquiera de las tres redes, es necesario trabajar los datos, eliminar los posibles registros que se hayan almacenado de manera incorrecta, eliminar las columnas innecesarias y escalar los datos para poder trabajar con ellos dentro del rango del 0 y el 1.

Todas las redes implementadas poseen una entrada de 720 registros, donde cada uno de los registros tiene 8 variables que son, el RSSI, ángulo de azimuth, ángulo de elevación, canal, altura de la antena, distancia entre la antena y el tag, altura del tag y el error medido valores obtenidos del archivo csv.

La primera red es completamente lineal, La primera capa oculta es de 64 neuronas, la segunda también posee 64 neuronas, entre la primera y la segunda capa tenemos una capa con Dropout de un 40% para evitar el overfitting . Finalmente, tenemos una neurona a la salida, se utilizó un optimizador Adam con un tamaño de paso de 0.01 y 20 épocas de entrenamiento con un batch size de 64. Los resultados obtenidos son los siguientes

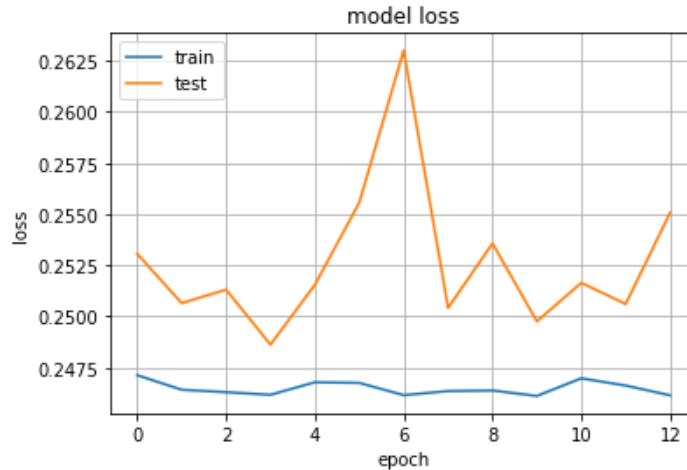


Figura 3.12. Error de la red neuronal lineal

El gráfico del error durante el entrenamiento y con los datos de prueba. Se observa que el modelo no aprende bien, esto debido al caso de tener un sistema completamente lineal, esto es equivalente a usar una neurona. Aumentar las épocas de entrenamiento tampoco es una solución, puesto que los resultados obtenidos no mejoran en lo absoluto, se probó cambiar la estructura de la red, pero no hubo mejoras. Se realizaron pruebas con la red de todas formas para tener un patrón de comparación.

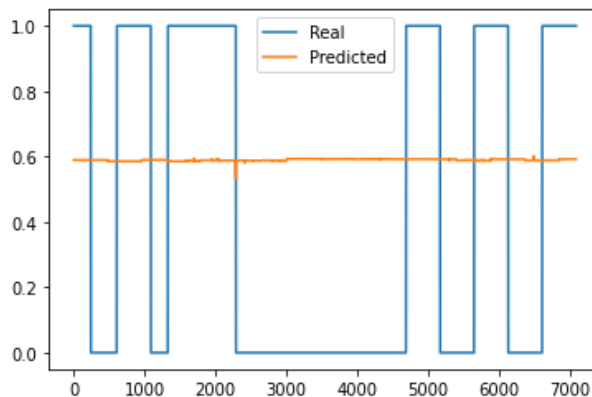


Figura 3.13. Gráfico de predicciones vs los valores reales obtenidos por la red neuronal lineal

La segunda red se implementaron funciones de activación, la primera capa oculta posee 128 neuronas, la segunda capa oculta posee 64 neuronas y una función de activación relu, se añade una capa de dropout al 40%, en la tercera capa tenemos 128 neuronas y una función de activación relu, y la capa de salida posee una neurona con la función de activación sigmoid. Se utilizó nuevamente el optimizador Adam con un tamaño de paso de 0.01, se realizaron 20 épocas de entrenamiento con un batch size de 64. Se obtuvieron los siguientes resultados.

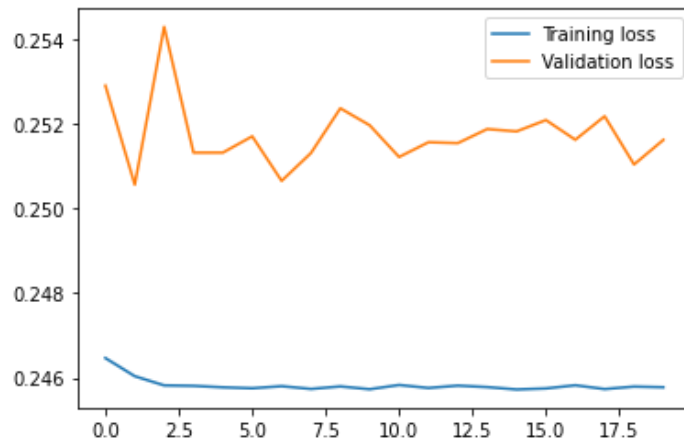


Figura 3.14. Error en el entrenamiento de la red neuronal no lineal

Ante este gráfico podríamos pensar que la solución, es aumentar el número de épocas de entrenamiento, se hicieron pruebas con 100 épocas y el resultado no varía demasiado, las perdidas por entrenamiento decrecen muy poco y al usar los datos de validación conserva de igual manera ese comportamiento de aumento y disminución. Se realizo de igual manera el gráfico de los valores predichos contra los valores reales, se observa muy poco cambio con respecto a la red neuronal lineal.

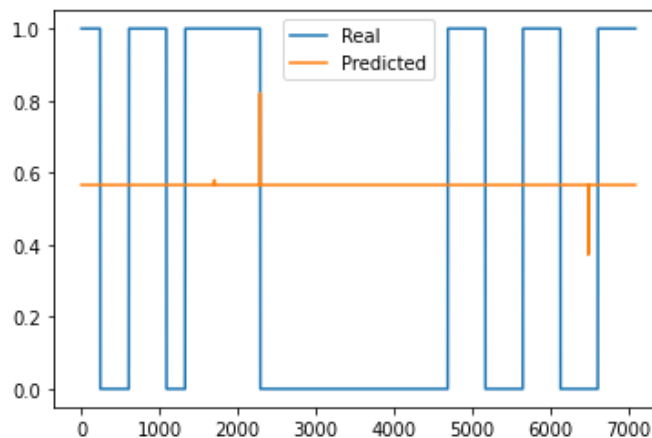


Figura 3.15. Gráfico de predicciones vs los valores reales obtenidos por la red neuronal no lineal

Para la tercera red se implementa una red neuronal convolucional, esta posee una capa convolucional con cuatro filtros, y el tamaño del kernel es de 5 por 5, esta capa posee una función de activación relu, la siguiente capa es una de max pooling de 2 por 2 , la tercera capa es una compuesta por una red Flatten, la cuarta es una red densa con 32 neuronas y función de activación relu, una capa de dropout de un 50% y una capa de salida de 1 neurona con la función de activación sigmoid.

Se escogió un optimizador sgd, nuevamente se usaron 20 épocas de entrenamiento y el tamaño del batch size es de 64.

Los resultados obtenidos son los siguiente.

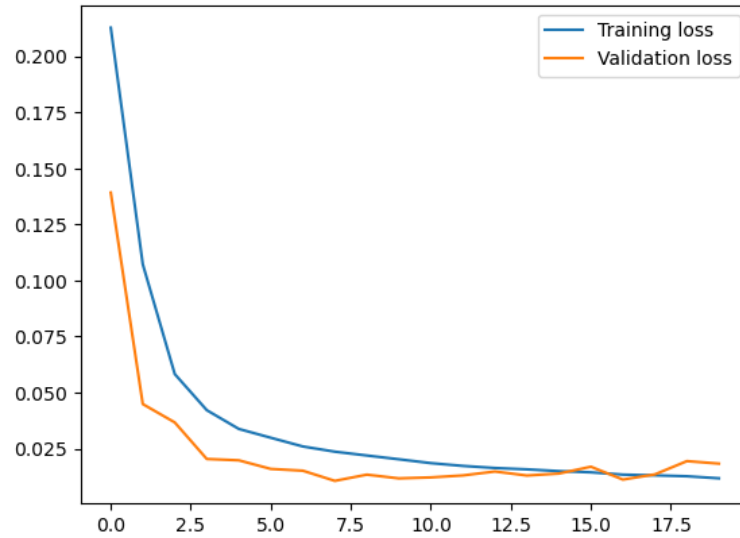


Figura 3.16. Error en el entrenamiento de la red neuronal convolucional

Podemos observar que esta red está aprendiendo los patrones de los datos, incluso se ve que con esta configuración y con el dropout determinados por el ordenador, no serían necesarias mas de 10 épocas de entrenamiento, una vez pasa esa cantidad la red estaría aprendiendo el ruido de los datos. Se realiza el gráfico de las predicciones contra el valor real como se puede ver en la figura 3.17, se observa que el modelo es capaz de predecir bastante bien. De igual manera se pueden mejorar aún más, puesto que no existen datos intermedios, está la línea de visión libre o no lo está.

Por lo que se puede implementar un umbral que permita tener valores desde el 1 hasta el 0.

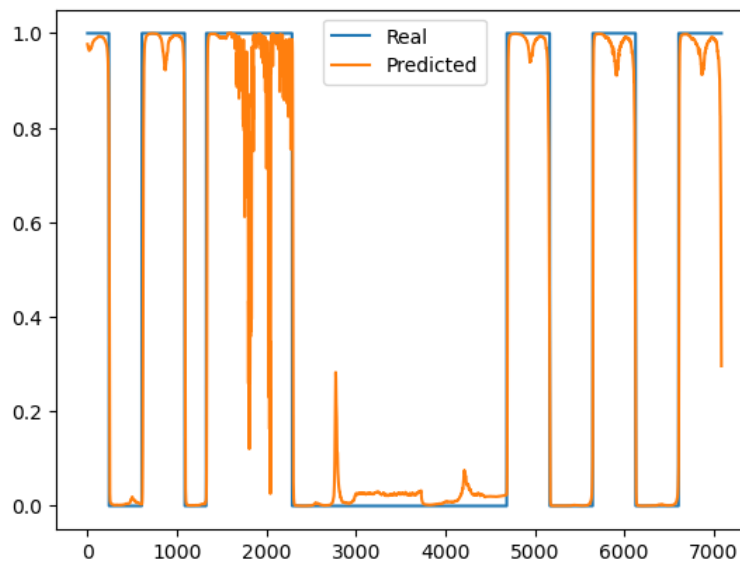


Figura 3.17. Gráfico de predicciones vs los valores reales obtenidos por la red neuronal convolucional

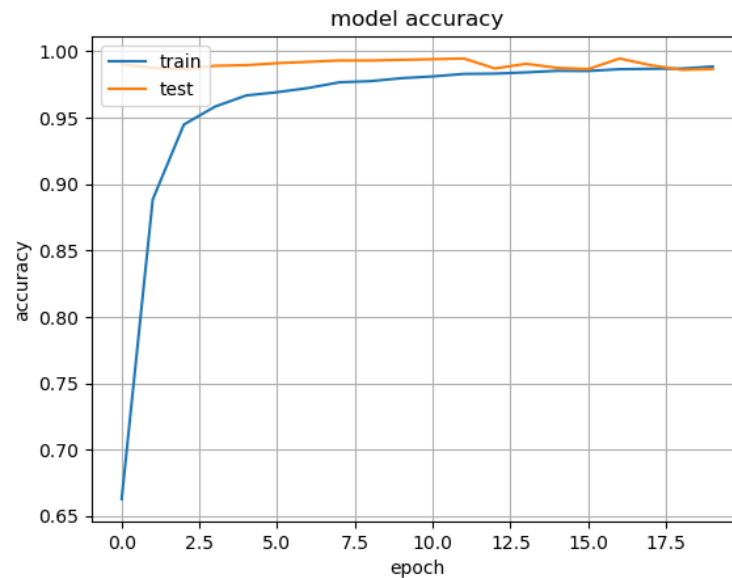


Figura 3.18. Gráfico de la precisión del modelo convolucional

Tenemos una excelente precisión en los primeros instantes, es posible que como se mencionó anteriormente exista un sobre entrenamiento por parte de la red.

Quedan pendiente por realizar las siguientes tareas

- Completar la teoría de las arquitecturas de redes neuronales junto con los tipos de aprendizajes.
- Realizar el sumario, la metodología, dedicatoria, agradecimientos.
- Aumentar la revisión bibliográfica.
- Aumentar el dataset.
- Seguir realizando pruebas con la red neuronal convolucional

Las siguientes tareas que debo comenzar a realizar

- Montar un servidor local para poder enviar los datos directamente al servidor.
- Implementar una interfaz que nos permita visualizar el tag en movimiento.
- Realizar la red neuronal LSTM.
- Seguir documentándome.

Referencias

- [2] P. Spachos and K. Plataniotis, "BLE Beacons in the Smart City: Applications, Challenges, and Research Opportunities," in *IEEE Internet of Things Magazine*, vol. 3, no. 1, pp. 14-18, Mar. 2020, doi: 10.1109/IOTM.0001.1900073.
- [3] M. Woolley. "Bluetooth Core Specification v5.1, Feature Overview," *Bluetooth Special Interest Group*, version 1.0.1, Dec. 2020.
- [3] P. Karlsson. "Getting started with Bluetooth for high precision indoor positioning", *U-blox AG*, Sep. 2021.
- [4] M. Woolley. "Bluetooth Direction Finding, A Technical Overview," *Bluetooth Special Interest Group*, version 1.0.3, Oct. 2021.
- [5] M. Amer and K. Atteya. "Indoor Positioning Bluetooth Angle of Arrival," M.S thesis, Department of Electrical and Information Technology, Lund University, Sweden, 2019
- [6] M. Ji, J. Kim, J. Jeon and Y. Cho, "Analysis of positioning accuracy corresponding to the number of BLE beacons in indoor positioning system," 2015 17th International Conference on Advanced Communication Technology (ICACT), 2015, pp. 92-95, doi: 10.1109/ICACT.2015.7224764.
- [7] A. Noertjahyana, I. A. Wijayanto and J. Andjarwirawan, "Development of Mobile Indoor Positioning System Application Using Android and Bluetooth Low Energy with Trilateration Method," 2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIT), 2017, pp. 185-189, doi: 10.1109/ICSIT.2017.64.
- [8] P. Andersson and L. Persson, "Evaluation of Bluetooth 5.1 as an Indoor Positioning System," Dissertation, 2020.
- [9] P. Sambu and M. Won, "An Experimental Study on Direction Finding of Bluetooth 5.1: Indoor vs Outdoor," 2022 IEEE Wireless Communications and Networking Conference (WCNC), 2022, pp. 1934-1939, doi: 10.1109/WCNC51071.2022.9771930.
- [10] P. Babakhani, T. Merk, M. Mahlig, I. Sarris, D. Kalogiros and P. Karlsson, "Bluetooth Direction Finding using Recurrent Neural Network," 2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2021, pp. 1-7, doi: 10.1109/IPIN51156.2021.9662611.
- [11] Y. Shi, Y. Long, F. Lu, Z. Xu, X. Xiao and S. Shi, "Indoor RSSI Trilateral Algorithm Considering Piecewise and Space-Scene," 2017 IEEE International Conference on Smart Cloud (SmartCloud), 2017, pp. 278-282, doi: 10.1109/SmartCloud.2017.52. [3].

Departamento	: Departamento de Ingeniería Eléctrica
Carrera	: Ingeniería civil electrónica
Nombre del memorista	: Jocsan Magdiel Vergara Blanco
Título de la memoria	: Título de la memoria
Fecha de la presentación oral	: Día/mes/año
Profesor(es) guía	: Nombres
Profesor(es) revisor(es)	: Nombres
Concepto	: Aprobado/Bueno/Muy bueno/Sobresaliente
Calificación	: Nota

Resumen (máximo 200 palabras)