Channel State Information based Indoor Localization using Convolutional Neural Network

Stefanos Stergiopoulos

SID: 3308210040

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

APRIL 2023

THESSALONIKI – GREECE

Channel State Information based Indoor Localization using Convolutional Neural Network

Stefanos Stergiopoulos

SID: 3308210040

Supervisor:	Prof. Stavros Stavrinides
Supervising Committee	Dr. Leonidas Akritidis
Members:	Asc. Prof. Christos Tjortjis

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

Master of Science (MSc) in Data Science

APRIL 2023

THESSALONIKI – GREECE

Abstract

Indoor positioning has attracted scientific interest due to the need to improve quality of life through advanced human-centric services. GPS technology is unable to cover indoor environments, thus modern indoor localization techniques can overcome this issue. Among others, fingerprinting-based methods are supposed to be highly promising due to their high accuracy and low requirement for equipment. The fingerprinting approach refers to the procedure in which signals are collected at known locations in an experimental area and, afterwards, estimation of the locations of previously unknown incoming signals takes place. The database with the collected fingerprints plays a crucial role in the process of comparing new incoming signals with existing ones and can be used to distinguish distinct positions clearly. This approach of fingerprinting interests many researchers due to its applicability with machine learning implementations.

Earlier, most methods were based on Received Signal Strength Indicator (RSSI) due to its low cost and complexity, however, it is vulnerable to noise signals, multipath and Non-Line-of-Sight (NLOS) in contrast to Channel State Information (CSI). In this research, CSI fingerprinting method is used to estimate the location of a device in an indoor environment. To achieve this, two devices are used as receivers and are placed diagonally in two of the corners of the experimental room while one more device is used as the transmitter and is placed in 6 locations in the room. Each device consists of one antenna. The CSI information was collected by the company Ariadne Maps GmbH which is located in Munich, Germany, and the devices used for this purpose were the sensors ESP32 from ESPRESSIF.

The amplitude and phase of the CSI are preprocessed before being used to reduce noise and achieve better results. A single fingerprint contains both amplitude and phase information which are the first and the second channel of a CSI image input to a Convolutional Neural Network. The average Euclidean distance between actual and predicted locations is the criterion chosen for measuring the performance of the model while experimental results show that the proposed algorithm exhibits low average distance error. The performance of the model is compared for the cases of using only amplitude, only phase and finally both amplitude and phase as the input image to the CNN.

Acknowledgments

First of all, I would like to thank and express my deepest gratitude to my supervisor, Prof. Stavros Stravrinides for the time he spent giving me advice, guidance and strong support throughout this academic journey.

Furthermore, I must express my gratefulness to the company Ariadne Maps GmbH for providing professional guidance and the CSI data and also the PhD student Anastasios Foliadis for his suggestions and technical support.

I would also like to express my respect and gratitude to my dissertation committee members, Prof. Christos Tjortjis and Prof. Leonidas Akritidis.

Stefanos Stergiopoulos

07/04/23

Contents

List of Tables

List of Figures

List of Acronyms and Abbreviations

Global Positioning System (GPS)

Non-Line of Sight (NLoS)

Line of Sight (LoS)

Orthogonal Frequency Division Multiplexing (OFDM)

Localization-Based Services (LBS)

Angle of Arrival (AoA)

Channel State Information (CSI)

Received Signal Strength Indicator (RSSI)

Channel Frequency Response (CFR)

Medium Access Control (MAC)

Radio Frequency (RF)

Time Difference of Arrival (TDoA)

Indoor Positioning Systems (IPS)

Time of Arrival (ToA)

Received Signal Strength (RSS)

Physical layer (PHY)

Radio-Frequency Identification (RFID)

Ultra-Wide Band (UWB)

Multiple Inputs Multiple Outputs (MIMO)

Infrared Radiation (IR)

Angle of Arrival (AoA)

Inertial Measurement Unit (IMU)

Light-Emitting Diode (LED)

Time of Flight (ToF)

K-Nearest Neighbor (KNN)

Support Vector Machine (SVM)

Bluetooth Low Energy (BLE)

Ultra-Wide Band (UWB)

Time Difference of Arrival (TDoA)

Access Point (AP)

Principal Component Analysis (PCA)

K-Nearest Neighbors (KNN)

Inverse Fourier Transform (IFT)

Channel Impulse Response (CIR)

Long Short-Term Memory networks (LSTM)

Multi-Layer Perceptron (MLP)

Artificial Neural Network (ANN)

Restricted Boltzmann Machine (RBM)

Convolutional Neural Network (CNN)

1 Introduction

The rapid growth of mobile devices and development of wireless technologies have shown the path for human-centric services. Consequently, Location Based Services (LBS) have rapidly become an extremely interesting and attractive field for both research and industry.

One of the available choices LBS provide is to determine the location of a person or an object which is referred to as localization. Localization can be distinguished in self-localization and aided localization [1]. The first term is used when a person uses their natural abilities to detect their location while aided localization term is used when electronic devices are used to obtain user's current location.

Aided localization can be further divided into indoor and outdoor positioning. As it is profound, outdoor positioning occurs in an outdoor environment, namely outside buildings. On the other hand, indoor positioning is performed indoors, inside several types of buildings the application of which could be useful, such as shopping malls, hospitals, airports and houses among others.

For outdoor localization the main satellite positioning systems include the Global Positioning System (GPS) developed by the United States Navy, the Global Navigation Satellite System (GLONASS) developed by the Soviet Union, BeiDou Navigation Satellite System developed in China and GALILEO which is more popular in Europe and was created by the European Union.

Among the main satellite positioning systems, the most popular is the GPS, which has impressive performance outdoors, but that does not occur with Indoor Positioning Systems (IPS) which deal with more challenges. GPS can be found in several outdoor applications such as in military field, agriculture, mapping and navigation, etc. The location and velocity of the tracked object can be determined using GPS signal receivers which are installed on the devices and receive the signals emitted from the satellites. Location estimation varies from 2 to 6 meters during normal operation [2].

However, this technology cannot be applied in indoor environment due to Non-Line of Sight, since GPS signals are not able to pass through rooftops, walls or obstacles which are present in a complex indoor environment. Besides, in indoor environments greater accuracy is required which could not be achieved with GPS even if the satellite signal blockage was not a problem.

The continuous development of wireless technology and the widespread adoption of smartphones and generally the *Internet of Things* have resulted in the development of a wide variety of services. These services include LBS such as IPS, which refers to the procedure of estimating an object's location or user in an indoor environment. Usually, an indoor environment consists of many rooms, corridors with complex arrangements, solid walls, several furniture and people doing several activities or simply sitting, standing or walking. As a result, electromagnetic signals are attenuated, reflected and suffer from interference, multipath, fading and delay phenomena.

IPS refers to the system used to localize people or objects indoors. The information of the location is sent to the system which makes sense of it and provides the location estimation. The design of the IPS depends on the type of indoor positioning technologies, algorithms and techniques used in the system. The are several technologies used in IPS, some of which are the Radio Frequency Based technologies such as Ultrawideband, Bluetooth Low Energy and Wi-Fi, Satellite-Based, Inertial Sensor-Based and Optical-Based. For each application the appropriate selection of suitable technology is required in order to fit the special need and increase performance while reducing cost.

1.1 Thesis outline

The outline of this research is as follows. First, Chapter 2 contains a review about indoor localization where its applications are presented and fingerprinting using Wi-Fi is explained. Chapter 3 presents the related work. In Chapter 4, technical background about localization techniques and technologies used in indoor localization are introduced, while Chapter 5 contains principles of Neural Networks. The experimental setup and the results are examined in Chapter 6 while, finally, Chapter 7 presents the conclusions and future work.

2 Indoor localization

2.1 Applications of indoor localization

The growth of indoor localization and the widespread development of smart portable gadgets has permitted various location-based services. Briefly, some useful applications of LBS indoors could be the following.

For example, in a supermarket the LBS could navigate the customer to the product they are interested in. The cart could be equipped with a screen which contains a list of the products that are available in the supermarket and using a combination of technologies such as Wi-Fi and Radio Frequency Identification tags, the customer can get directions to the target products. Also, the marketing team can promote advertising campaigns and send exclusive offers to the customers. A breakthrough concept in e-commerce, contextual-aware location-based marketing has the potential to increase sales and profitability. This kind of advertising enables the merchant to interact with customers immediately and improve their purchasing experience. This is significant, particularly given recent technological advancements and the prevalence of smart mobile devices. Widespread use of customized mobile devices enables targeted marketing strategies based on consumers' location, social media profiles, spending habits, navigation history, online behavior, browsing patterns, and preferences. The purpose of this marketing technique is to make assumptions about the consumer's particular interests, previous purchasing behavior, requested comments, and email reminders before sending them pertinent adverts and coupons from stores nearby.

Another interesting application is the use of LBS in museums or art galleries. The visitor, instead of using brochures, can be equipped with a device which provides navigation services and information about the pieces of art available. They can select the piece of art they want to see, while information about it can be given through headphones simultaneously or take a complete tour in the museum on their own using their smartphone as a personal guide. Also, the device could provide the visitors with real-time information about pieces with high congestion to avoid them, and inform them, when is the appropriate time to visit them.

Similar to the use of LBS in a supermarket, such an application could also be useful in a library. People who visit a public library or students at a university's library could install an application on their smartphone, search for the book they are interested in, in the application's database and get directions to reach the book. Of course, the app may navigate the user with some meters of accuracy, but it leads them remarkably close to the target book and saves time from searching the whole library.

In healthcare domain, indoor localization can surprisingly improve the services provided. In a hospital localization of patients who suffer from mental diseases could protect them from being hurt. Patients with Alzheimer may get lost in the hospital and reach a place far away from their room or even other floors. Hospital personnel could be able to locate them immediately and direct them back to safety, or other wandering patients can be located and receive their treatment on time. Visitors to the hospital may have little trouble locating their patients. Even wheelchairs and sophisticated surgical equipment are readily available within operating rooms.

In emergency situations where people were trapped indoors after an earthquake or a fire, indoor localization techniques may pinpoint the precise location of those who were in danger and quickly remove them from the structure while saving valuable time in detecting survivors. Accounting for the precise number of people that are trapped and rescuing them safely could be challenging because the inside environment is typically unknown to the rescue team. The ideal tool for a rescue force may be a positioning system that does not require prior measurement, calibration or configuration.

In extreme circumstances, the disaster may also cause the built-in communication system to fail. In this kind of situation, context-aware positioning can completely transform the game.

In more simple applications such as smart homes, LBS may adjust the temperature according to the current location of the users and their daily behaviors. In the industry domain, people and asset tracking can be applied in the *Industry 4.0* framework while warehouse monitoring and autonomous robot navigation may also be implemented.

LBS does not have to do only with people. There are situations where it is necessary to track several objects automatically in sizable factories and warehouses. In this situation, controlling the locations of these items in real time is necessary in addition to localization and identification. To ensure that all these things are correctly identified without collisions and blockages, new Medium Access Control layer protocols are also required in addition to localization mechanisms and database management. To handle that enormous amount of data, deep learning techniques are frequently required.

Figure 2.1 presents an overview of indoor and outdoor LBS, from which indoor applications can take part in mall, prison, factory, transportation, community, education center and hospital.

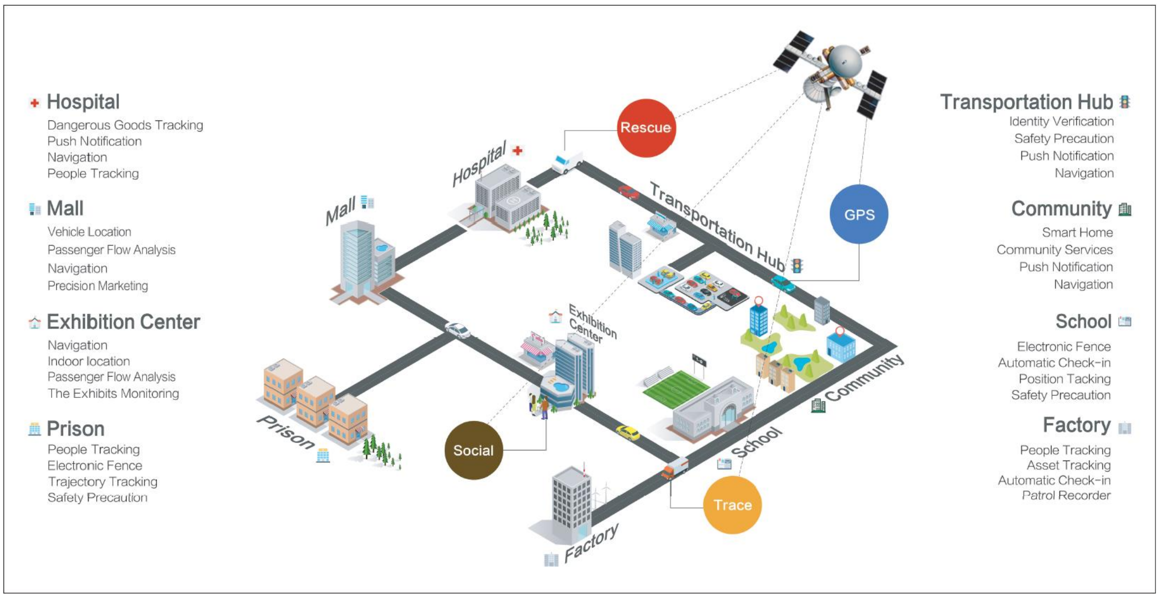


Figure 2.1 An overview of indoor and outdoor Location Based Services [3]

2.2 Indoor localization using Wi-Fi

Wi-Fi is among the most widely utilized technologies for indoor localization. The reason for that is that the number of devices which support Wi-Fi are used by people is increasing daily. Furthermore, Wi-Fi access points are already installed in almost all buildings to provide Internet access like shopping malls, airports, hospitals, office buildings and others. As a result, no additional infrastructure is needed and that makes Wi-Fi technology suitable for indoor LBS which is supposed to be applied in promising systems. However, accurate indoor location estimations, at a sub-meter lever is still a challenging research issue.

2.2.1 Geometric mapping and fingerprinting

There are two main approaches for indoor localization using Wi-Fi signals. These are geometric mapping and feature pattern mapping, or fingerprinting, as it is widely known. Geometric mapping approach is based on the calculation of geometric parameters like signal power, distance and angle with respect to reference points [4]. Then follows the process of location estimation using geometric algorithms such as triangulation, trilateration or multilateration. One main issue with that approach is the irregularities in the signals due to multipath effects. To deal with that issue, fingerprinting approach builds a database with feature patterns obtained from the Wi-Fi signals. Each location has its own pattern. In that way, localization has turned into a problem of comparing and detecting the best matched pattern for the received signal at the unknown location from the patterns available in the database. The complex indoor environment affects the propagation of the Wi-Fi signals and as a result each location in the examined area has its own signal pattern which is known as the Wi-Fi fingerprint. The more complex the indoor environment, the easier distinguishable the fingerprint, and as a result each location, will be. The IPS take advantage of that characteristic of Wi-Fi fingerprinting to perform accurate location estimations.

One main drawback of the geometric mapping approach is that the calculation of distance and direction information depends strictly on the Line-of-Sight conditions. Line of Sight conditions in a complex indoor environment is almost always hard to obtain. Fingerprinting deals better with such issues because the feature pattern does not depend on LOS scenarios.

2.3 Wi-Fi fingerprinting

Wi-Fi fingerprinting is a famous approach in IPS which is based on Wi-Fi technology. An example of a fingerprinting based indoor localization schema is depicted in Figure 2.2.

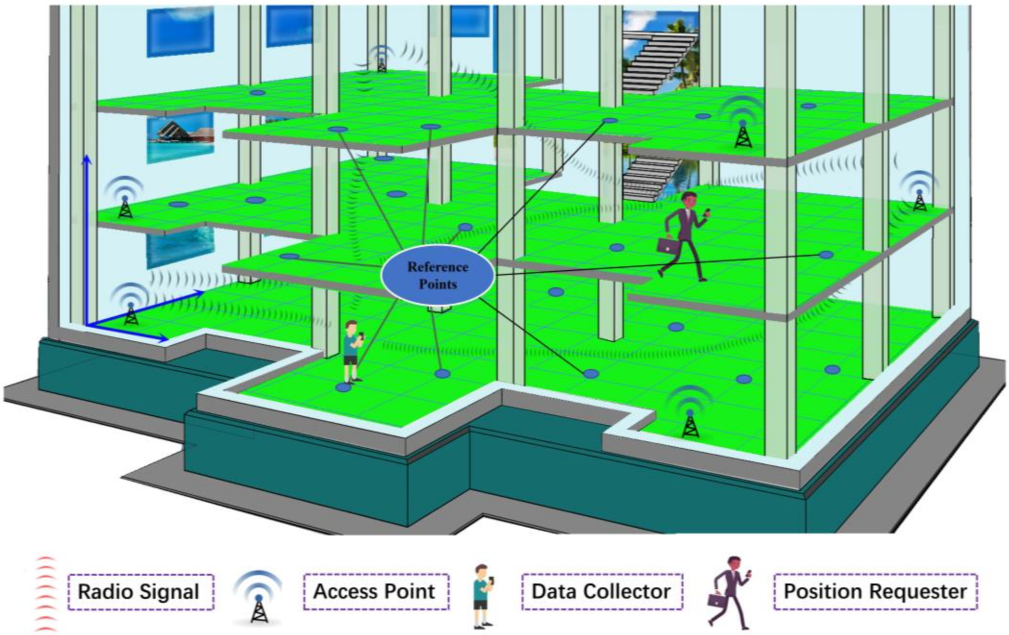


Figure 2.2 Indoor localization based on fingerprinting [5]

There are two phases involved in traditional fingerprinting, the offline and the online phase. During the offline phase, a database is constructed with the signal characteristics of each predetermined reference point collected as fingerprints. These fingerprints contain the RSSI or CSI values and the coordinates of the reference point's location. Preprocessing techniques on the RSSI or CSI values may be useful to reduce the noise. During the online phase the newly obtained signal characteristics from single or several access points, which are the fingerprint at the unknown location, are matched with the database information. The new location is estimated to be the one with the highest matching degree from those contained in the database in terms of signal characteristics.

In fingerprinting, there are two approaches to estimate the location of the user or an object. The first one is to divide the space in which we want to perform localization into small subregions and perform a classification task using machine learning classifiers. The machine learning model will determine the subregion to which the user belongs. In that method, the size of the subregions affects how accurate the model is. The second approach is to interpolate a location from the locations contained in the database. In many applications this was done using a deep artificial neural network or a convolutional neural network. However, with this approach the inaccuracy increases significantly as we move to locations that are not contained in the database.

There have been proposed many techniques for localization, but fingerprinting is supposed to be the most promising one since it provides greater accuracy. Even though the implementation is quite troublous and time consuming, the operation is less complex when compared to other techniques. Furthermore, in fingerprinting technique existing infrastructure can be used. However, there are still some issues some of which are depicted in Figure 2.3.

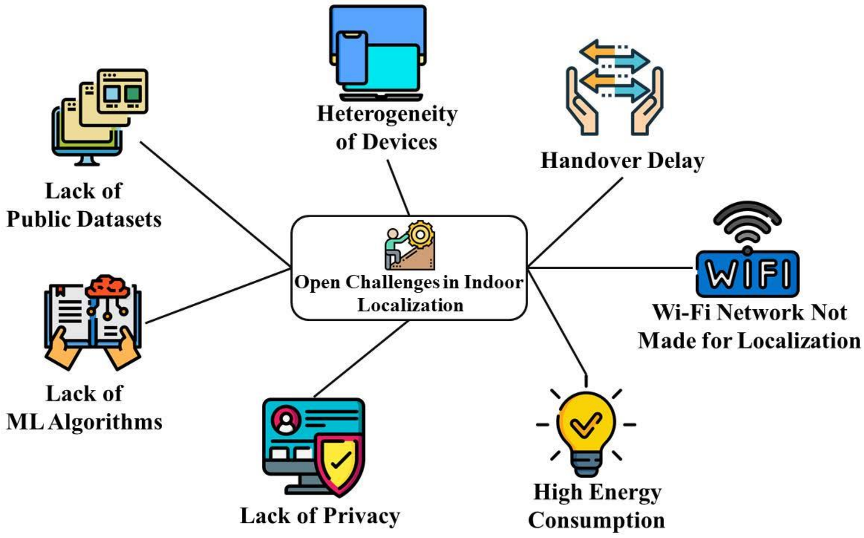


Figure 2.3 Open challenges of fingerprinting-based indoor localization [6]

2.4 RSSI vs CSI

Wireless measurements such as RSSI and CSI are used in fingerprinting-based systems. RSSI refers to the strength of the received signal and belongs to the MAC layer, while CSI belongs to the physical layer (PHY) of the wireless communication protocol. RSSI is the aggregation of all the signals that travelled across multiple paths. Changes in the environment affect the signal attenuation and reflection and as a result the RSSI values may fluctuate [7]. One drawback of using RSSI for indoor localization is that it depends on LOS while on the other side CSI, which is considered an enhanced descriptor of wireless propagation, can characterize multipath effects in NLOS. Besides, NLOS is desired in fingerprinting because more unique patterns will be created.

CSI has two forms. These are Channel Impulse Response (CIR) and Channel Frequency Response (CFR). CIR represents the complex channel in time domain and CFR in frequency subcarriers. CFR can be easily obtained over devices that use Orthogonal Frequency Division Multiplexing (OFDM). The CIR type is more vulnerable to errors because of poor synchronization between devices, while in CFR phase calibration techniques can deal better with those synchronization issues.

Before using RSSI or CSI data in their raw form, preprocessing before using them in the localization algorithms, may be necessary to improve localization. Raw RSSI values gathered from multiple access points may undergo preprocessing to generate radio maps for use in clustering or extract statistical metrics (e.g. median, standard deviation, mean) as the fingerprints. In case of CFR, a unique fingerprint at each location can be generated by subtracting the mean CFR amplitude and phase value from the original CFR or by preserving only the steadier set of subcarriers, considering that each subcarrier undergoes different fading.

2.5 Machine Learning benefits in indoor localization

One challenge that localization algorithms must deal with is the volume of data needed to be processed. In some applications the signals of an extremely high number of access points need to be analyzed to obtain location estimation of the user or a device as accurate as possible. Traditional machine learning techniques find it difficult to deal with that data which finally leads to a time-consuming system. Nevertheless, deep learning modeling is already used in recent systems and makes the procedure much slower when dealing with massive volumes of data while also providing more accurate location estimations. Deep learning models can be used either for classification or regression for indoor localization purposes.

The presence of high dimensional data is a barrier to the development of localization systems based on fingerprinting. It is possible to use dimension reduction techniques, such as Principal Component Analysis, to transform high-dimensional data into low-dimensional data. The necessary storage and the computational complexity will be significantly reduced. Figure 2.4 illustrates the process of Machine Learning based indoor localization using Wi-Fi fingerprints.

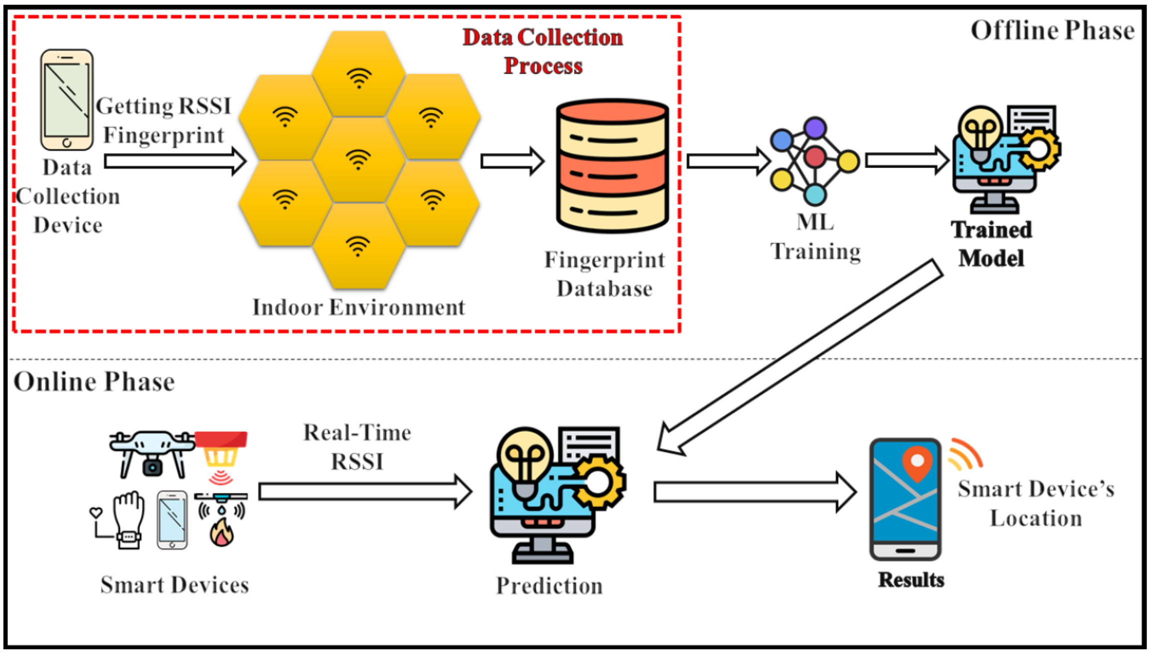


Figure 2.4 Indoor Localization using Wi-Fi fingerprints based on ML [6]

A promising branch of Machine Learning is Reinforcement Learning (RL) that can be used in robot navigation. RL can help robots construct an adaptive control system which is trained through experiences.

Two especially important evaluation metrics in indoor positioning systems are scalability and adaptability. The system needs to be designed in a way that allows for simple environmental change adaptation. Transfer learning can play a significant role as it can help machine learning models to be trained faster in unfamiliar environments by making comparisons with what was previously learned.

3 Related Work

This chapter presents some background about indoor localization mainly based on RSSI and CSI.

Localization algorithms based on fingerprinting and RSSI values are broadly used. The main drawback of using RSSI is the restriction of LOS scenarios which is necessary in localization based on RSSI. To deal with RSSI fluctuations, the authors in [8] achieve localization with a room level accuracy using SVM.

In [9] a Recurrent Neural Network was developed that uses the trajectory and RSSI information. Several types of RNNs were used such as Vanilla RNN, Gated Recurrent Unit, LSTM and some bidirectional types of RNNs.

A technique called DeepFi has been created by the authors of [10] and [11] to enhance the effectiveness of deep learning localization based on fingerprinting. Their approach extracts the CSI amplitude from the available subcarriers and, during the offline training phase, trains a deep network using a stack of Restricted Boltzmann Machines. They employed Radial Basis Function to achieve localization during the online phase.

Authors in [12] developed a localization method using a CNN and Wi-Fi, called CiFi. Their method gathers CSI data and uses the phase data to calculate AoA. After that they construct the CSI images using the calculated AoA and use them as the input of the CNN during the offline phase. The new location is estimated using the trained CNN and the newly obtained AoA in online phase.

In [13] a multi-layer perceptron and a 1D-CNN were used to achieve localization. The results indicate that, regardless of the low complexity of the used CNN, the algorithm reaches great localization accuracy.

Authors in [14] investigate the application of fingerprint inputs to a Convolutional Neural network (CNN) for localization using channel state information (CSI). By considering the irregularities in its raw phase that render the CSI untrustworthy, they investigate if the CSI may be used as a distinct fingerprint corresponding to a single position. They suggest two techniques for creating trustworthy fingerprints that contain phase data.

Reference [15] presents FILA which is based on geometric mapping techniques. Localization is performed using CSI information and trilateration. The authors apply Inverse

Fast Fourier Transformation to convert the CSI data, get the channel response in time domain in order to reduce multipath effects.

A fingerprinting approach is presented in reference [16] called FIFS. A probability algorithm is used to estimate the location of the target after constructing a radio map from CSI information. The system works in two phases, namely calibration and positioning. In the first phase, it gathers CSI data at each location from three access points to obtain the fingerprint. A probabilistic approach is used in the positioning phase to locate the object. The algorithm achieves an acceptable mean error of 1 meter.

In [17] CSI-MIMO is presented which is a fingerprinting method that uses amplitude and phase of the CSI data. The radio map is constructed after aggregated CSI information from several antennae and subcarriers. The location estimation takes place with KNN and maximum likelihood estimation.

4 Technical Background

4.1 Localization Techniques

Localization techniques are several approaches that are used for localization purposes.

4.1.1 Received Signal Strength Indicator (RSSI)

One of the widely used approaches for indoor localization is Received Signal Strength Indicator (RSSI), well-known for its simplicity. RSSI, as its name reveals, is the signal power strength reaching the receiver. It is commonly measured in decibel-milliwatts (dBm) or milliWatts (mW). If there are two devices, one being the transmitter and the other the receiver, the RSS can be utilized to provide estimations about the distances between the two devices. The smaller the RSS value the bigger the distance between the devices since the strength of the signal deteriorates. This is known as the RSS's decay low with distance [18]. RSSI is the RSS indicator, a relative measurement of the RSS whose units are determined by each chip manufacturer. Atheros Wi-Fi chipset uses a range between 0 and 60 for the RSSI values and Cisco uses values between 0 and 100 [19]. An estimation about the distance between a transmitter and a receiver can be done with the following equation:

$$RSSI = -10nlog_{10}(d) + A$$

where n is the path loss exponent and A is the RSSI value at a reference distance from the receiver.

To achieve localization in a device-based localization scenario using RSS, trilateration is needed. This means that the absolute distance between the device and at least three, or N more broadly, reference points whose precise locations are known is estimated using the RSS at the device. After that, one can apply geometry principles to estimate the location of the device relative to the reference points. An example of that process is shown in the Figure 4.1. The way trilateration works is similar to GPS.

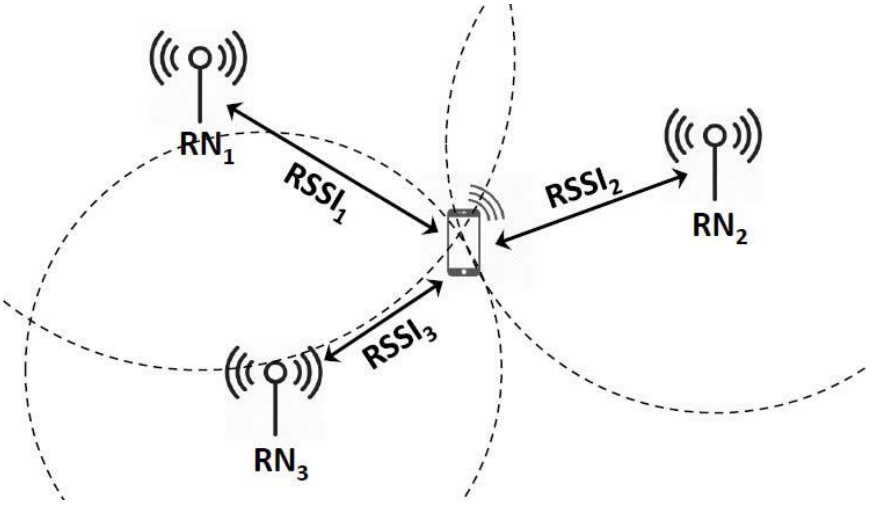


Figure 4.1 Received Signal Strength Indicator [19]

The RSS approach is popular thanks to its simplicity and cost-efficiency, but its drawback is that it achieves poor accuracy in localization. This is even worse in non-line-of-sight scenario. The insufficient localization accuracy happens because of the attenuation of the signal due to transmission through obstacles and the fluctuation of the RSS values caused by indoor environment noise and multipath fading. Fluctuations of the RSS values exist also due to human movements. Several filtering processes may improve the results, but acceptable accuracy is still unachievable.

Trilateration and fingerprinting are possible approaches to take advantage of the Wi-Fi RSS values for indoor localization purposes. Trilateration has already been discussed. Approximate perception is a simple approach which estimates the location of the device in accordance with the access point that has the highest RSS value.

Fingerprinting approach is the most popular among the previous two and is widely used in combination with machine learning. In indoor environments which are characterized with high complexity, Wi-Fi signals can be attenuated and reflected and as a result different location can accept RSS values from several transmitters. This technique uses the uniqueness, actually the fingerprint, which characterizes each location with specific RSS values. The role of the fingerprinting technique is to match the RSSI value of the unknown location with one from the database. On the Table 1 an example of a RSSI database is shown. The columns with index WAP001 to WAP520 contain the RSS values from multiple access points and the next columns represent the longitude and latitude, floor and building labels at the location at which the RSSI values were taken.

Table 1 Example of a RSSI database [20]

WAP001	WAP002	…	WAP520	LONGITUDE	LATITUDE	FLOOR	BUILDINGID
100	100	…	100	−7536.62	4864934.23	2	1
−75	−66	…	−93	−7510.44	4864949.25	3	2
100	100	…	−89	−7564.20	4864887.19	3	2
−58	100	…	100	−7336.70	4864764.48	2	1
100	−81	…	100	−7406.06	4864788.22	3	2
−77	100	…	−90	−7385.37	4864776.74	3	2
100	100	…	−92	−7391.08	4864779.91	3	2
−90	100	…	100	−7349.28	4864758.81	3	2
−74	−83	…	100	−7502.45	4864884.18	3	2
100	100	…	100	−7474.58	4864866.87	3	2

4.1.2 Channel State Information (CSI)

The Channel Impulse Response (CIR) or in frequency domain Channel Frequency Response (CFR) is delivered to upper layers as Channel State Information which provides rich information that can be obtained from Wi-Fi signals through Orthogonal Frequency Division Multiplexing (OFDM). CSI data belong to the physical layer (PHY) of 802.11 wireless communication protocol. This PHY of 802.11 protocol is the interface between MAC and wireless media [21].

CSI is more stable than RSSI in time but has strong spatial dependence. It is widely used in research to achieve sub-meter accuracy in indoor localization systems and contains two values which are amplitude and phase which can be used for the localization process as provides abundant information in the frequency domain. RSS can be easily obtained from the receiver while CSI data needs to be derived from the Wi-Fi driver of the device. Since it captures both amplitude and phase responses of the channel in different frequencies and multiple transmitter-receiver antennas pairs, it has higher granularity than RSS.

CSI is a complex quantity and can be written in polar form as:

$$H_k = |H_k| e^{j\angle \mathrm{H}_k}$$

where, $|H_k|$ is the amplitude response of the kth sub-carrier, or sometimes magnitude, and $\angle \mathrm{H}_k$ is the phase response of the frequency kth sub-carrier. H appears in a complex form $a + bi$ and we can get the amplitude by getting the modulus $\sqrt{(a^2 + b^2)}$ and the phase by getting $\theta = \alpha rgtan(\frac{b}{a})$.

Both range-based and range-free localization scenarios are compatible with this technique. [21].

4.1.3 Fingerprinting/Scene Analysis

Scene analysis localization techniques require fingerprint creation in the space where an object's location is estimated. It comprises of two phases. The offline and the online phase. In the offline phase, several RSSI or CSI values are collected at several locations in the space. Each location has its own measurement RSS or CSI values and in that way the fingerprint database is built. In the online phase, which is actually the real time estimation of an object's or user's location, the obtained measurements are matched to the values on the databases and the location is estimated. The are several algorithms used to fulfill such a task, some of those as discussed below.

4.1.3.1 Probabilistic methods

Probabilistic methods depend on the likelihood that an object or a user is at location x given the RSSI/CSI values which were gained during online phase. If we set that the possible locations at which a user may be, belong to the set:

$$L = \{L_1, L_2, L_3, \dots, L_m\}$$

and the vector of the RSSI or CSI values is O, then the user's location will be L_j, if:

$$P(L_j|O) > P(L_k|O) \; for \; j, k = 1, 2, 3, \dots, m \text{ with } k \neq j$$

The previous equation indicates that a user's location will be supposed to be L_j if its likelihood is higher than the likelihood being at any location L_k. If $P(L_j) = P(L_k)$ for $j, k = 1, 2, 3, \dots, m$, then we can obtain from Bayes' theorem the likelihood probability of the vector O, containing the RSSI or CSI values as $P(O|L_j)$. Finally, after mathematical operations, the user is supposed to be at location L_j if:

$$P(O|L_j) > P(O|L_k) \; for \; j, k = 1, 2, 3, \dots, m \text{ with } k \neq j$$

The RSSI or CSI measurements of the online phase are used in fingerprinting methods to map the user on a grid. Each spot in the space where the fingerprint database was constructed is associated to a point of the grid. Theoretically, it can be said that as the density of the grid is increased, the location estimation will be more accurate. Increasing the grid means increasing the number of observations in the fingerprint database. An issue is that as the distance between two spots is decreased, the difference of the signal strength between two neighboring spots, will also decrease and become much smaller than the signal fluctuation due to noise of the environment. That will make the differentiation of the two points almost impossible. So, a balance must be found between the fingerprinting grid density and the accuracy of location estimations. Furthermore, fingerprinting techniques are strongly dependent on changes in the environments on which the fingerprint database was obtained.

4.1.3.2 Neural Networks

Artificial Neural Networks are used in many problems where classification, regression or time-series forecasting take place. For localization, the Neural Network must be trained in the fingerprint database which contains the RSSI or CSI values and the user's location coordinates. The database construction and the training of the Neural Network is happening in offline phase. After training the NN, the user's location can be estimated after obtaining the new RSSI/CSI values online. The estimation of the location with a simple feed-forward Artificial Neural Network is obtained as follows briefly. First a vector which contains the RSSI/CSI measurements enters the network as the input layer, then is multiplied by the weights and the biases are added. Then it goes through the hidden layers and finally the output of the network is the estimation of the location.

4.1.4 Angle of Arrival (AoA)

This method calls for antenna arrays at the receiver in order to calculate the time difference of arrival at the individual antenna array components in order to determine the angle at which the signal sent from the transmitter reaches the receiver. Angle of Arrival approaches provide accurate results when the distance between the transmitter and the receiver is small, but its accuracy deteriorates when the distance is big. That is happening because a small error in the angle of arrival results in a huge error at the location estimation when the distance is long. Furthermore, these techniques require special hardware and accurate calibration when compared to other methods. Another issue is that in complex indoor environments where multipath effects are present, the Line-of-Sight scenario required for AoA approach is almost impossible [22]. These approaches use the angle and the distance measurement and as a result at least two anchor nodes with known locations are enough to estimate the position of the user [23]. For better accuracy more nodes can be used. The Figure 4.2 illustrates the way in which the user's location can be estimated with the use of AoA technique.

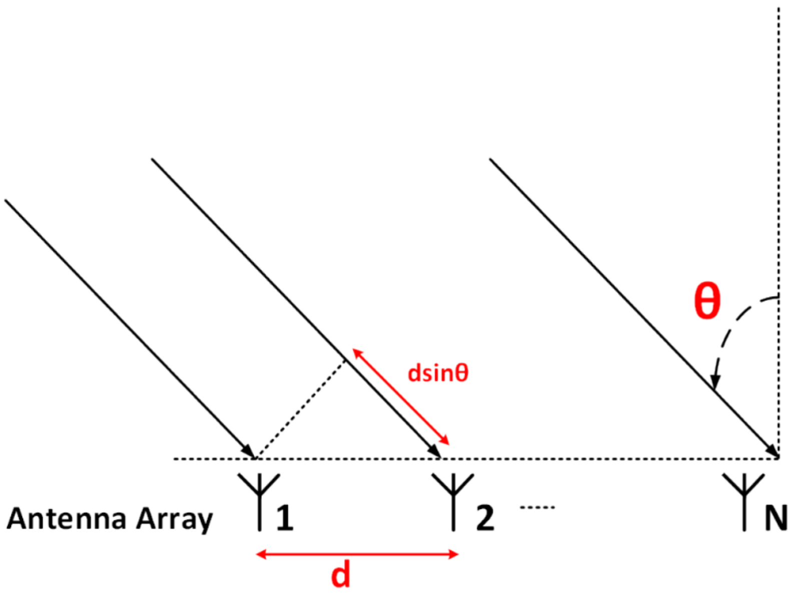


Figure 4.2 Angle of Arrival based localization [18]

4.1.5 Time of Flight (ToF)

The Time of Flight approach, as its name implies, calculates the exact distance between the transmitter and the receiver using the time it takes for the signal to travel between them. After exploiting it, it is multiplied by the speed of light, $c = 3x10^8 m/s$. Time of Flight is also called Time of Arrival. For rich results this method requires accurate synchronization between transmitter and receiver as a small difference can lead to extremely different values for the distance between the devices. The signal bandwidth and the sampling rate have a significant effect on ToF accuracy. If the sampling rate is low, there may be inaccuracies on the results since the signal may reach the receiver between the sampled intervals. However, even if large bandwidth and high sampling rate may increase the ToF accuracy, there are still localization errors due to non-line of sight scenarios in real-life applications. Due to deflections of the signals in obstacles or people in indoor environments, time travel increases, leading to an incorrect calculation for the distances. To acquire localization using this method at least three anchor nodes are needed, in a 2D environment as depicted in Figure 4.3 and four anchor nodes for 3D localization. Basic geometrical principles are used to estimate the location with respect to the three nodes.

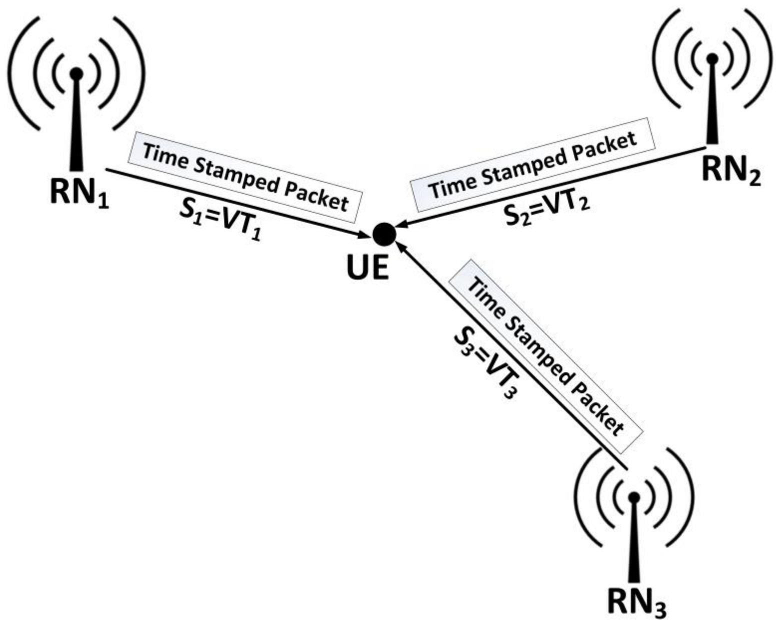


Figure 4.3 Time of Flight based localization [18]

4.1.6 Time Difference of Arrival (TDoA)

Time Difference of Arrival technique uses the difference in the propagation time of signals from multiple transmitters which reach the receiver. The Time Difference of Arrival measurement is defined as $T_{D(i,j)}$, which refers to transmitter i and j, is calculated with the speed of light $c = 3x10^8 m/s$, and the result is physical distance values $L_D = cT_{D(i,j)}$. The location of the receiver belongs to the hyperboloid expressed by the following equation:

$$L_{D(i,j)} = \sqrt{(X_i - x)^2 + (Y_i - y)^2 + (Z_i - z)^2} - \sqrt{\left(X_j - x\right)^2 + \left(Y_j - y\right)^2 + \left(Z_j - z\right)^2}$$

where (X_i, Y_i, Z_i) and (X_j, Y_j, Z_j) are the coordinates of transmitter i and j respectively and (x, y, z) are the coordinates of the receiver. Three or more transmitters are needed to estimate the location of the receiver as the intersection of the resulting hyperboloids after taking the TDoA as shown in Figure 4.4. This technique is different from the ToF where the absolute propagation time of the signal is used but the requirements for better accuracy are similar, namely the signal bandwidth and the sampling rate. The non-line of sight in indoor environments is still an issue.

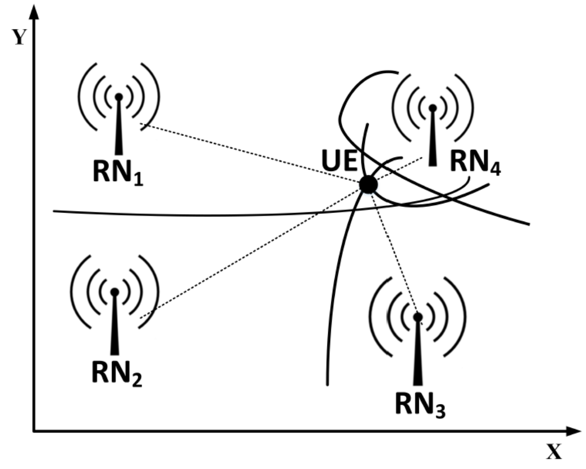


Figure 4.4 Time Difference of Arrival based localization [18]

4.2 Indoor Positioning Technologies

A vast variety of technologies are available for indoor localization purposes. Each one of those is better for a specific application. One should use the localization technology depending on their needs. The Figure 4.5 illustrates some examples of indoor positioning technologies.

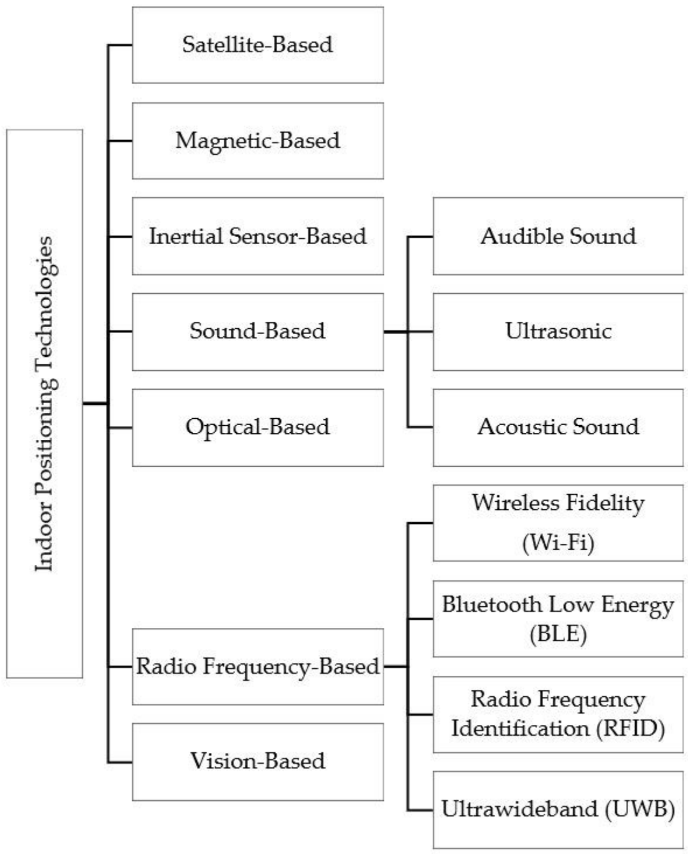


Figure 4.5 Types of Indoor Positioning Technologies [2]

4.2.1 Satellite-based

Satellite-based systems are the most popular in an outdoor setting. They take advantage of the network of satellites that communicate with radio signals with the earth. The Global Positioning System (GPS) provides the location estimation of the users with an error of less than 7.8 meters 95% of the time. These systems are easy to implement and cost-effective but are mainly functional in open space and not indoors where the signal deteriorates dramatically. GPS is used mainly in transport since its use is limited in manufacturing while it cannot be used in indoor environment.

4.2.2 Magnetic-based

There are two approaches in which magnetic fields can be used for indoor positioning purposes. The first one is to detect the anomalies caused by disturbances in the magnetic field of earth to perform localization. The other approach is to create manually magnetic fields. After the installation of coils which carry current, the measurement of each coil's magnetic field takes place.

The construction of an efficient magnetic field requires a map-building system. A data alignment system and odometry are included in the magnetic field. When the system is moving in the test area the position estimation is done using odometry.

4.2.3 Inertial Systems

Inertial Measurement Units or IMUs can be helpful in indoor localization as they can be used to estimate the location in reference to an anchor node. An IMU includes a gyroscope, an accelerometer and sometimes a magnetometer. These sensors can measure the rotational speed, linear accelerations and strength of the magnetic field and do not require external installation as they are independent of the environment. There are IMUs which also use an altimeter which measures the air pressure. The use of magnetometer in localization may lead to inaccurate results due to disturbances of indoor infrastructure.

4.2.4 Sound Based

This type of technology is mainly used in underwater applications such as monitoring and tracking. There are three subcategories include in sound-based technologies, audible sound, ultrasonic and acoustic sound. Audible sound properties are similar to those of ultrasound. Its application is cost-effective and is easier implemented for everyday use as in can be found in loudspeakers and smartphones. Ultrasound localization technologies mainly depend on Time of Flight techniques of ultrasound signals and use the velocity of sound to calculate the absolute distance between two devices which work as a transmitter and a receiver. This technology has proved to provide centimeter level accuracy. However, the velocity of sound depends strictly on humidity and temperature. On the other hand, ultrasound signals are slightly affected by complex indoor environments and have negligible penetration through walls. That fact makes ultrasound a promising technology in indoor localization systems. Acoustic signal-based technologies take advantage of the microphone sensors that are present in smartphones in order to detect acoustic signals transmitted by sound sources at a known location and estimation the position of the receiver in reference to these sources.

4.2.5 Optical Based

Optical -based localization is usually in the form of Electro-magnetic spectrum and can be divided into Infrared (IR) and Visible Light Communication (VLC). Infrared technologies can be found in a great variety of applications, from remote controllers to data transmission, since it is easily available and low-cost. In the literature is suggested that this technology can be used to provide real-time navigation to a cart in a supermarket. The VLC technology uses light sensors to detect the position and direction of the Light Emitting Diodes (LEDs). The LEDs work as the transmitters that emit the signal which is going to be used for localization purposes. Angle of Arrival technique achieves the highest accuracy in VLC technology. The location estimation is done using geometric properties of the triangle. The working principle of VLC is that each of the lamps used have different flickering encoding and the sensor, which could be a camera of a smartphone, compares the received encoding scheme with the known ones, selects the most dominant and corresponds the location of the sensor with that of the lamp. An example can be found in Figure 4.6. One main drawback is that Line of Sight is required to achieve localization. VLC has been considered for indoor positioning because it takes advantage of the already installed infrastructures and minimizes the extra implementation costs. Furthermore, it does not affect the indoor environment aesthetically since it can use conventional lamps installed in standard places as the ceiling.

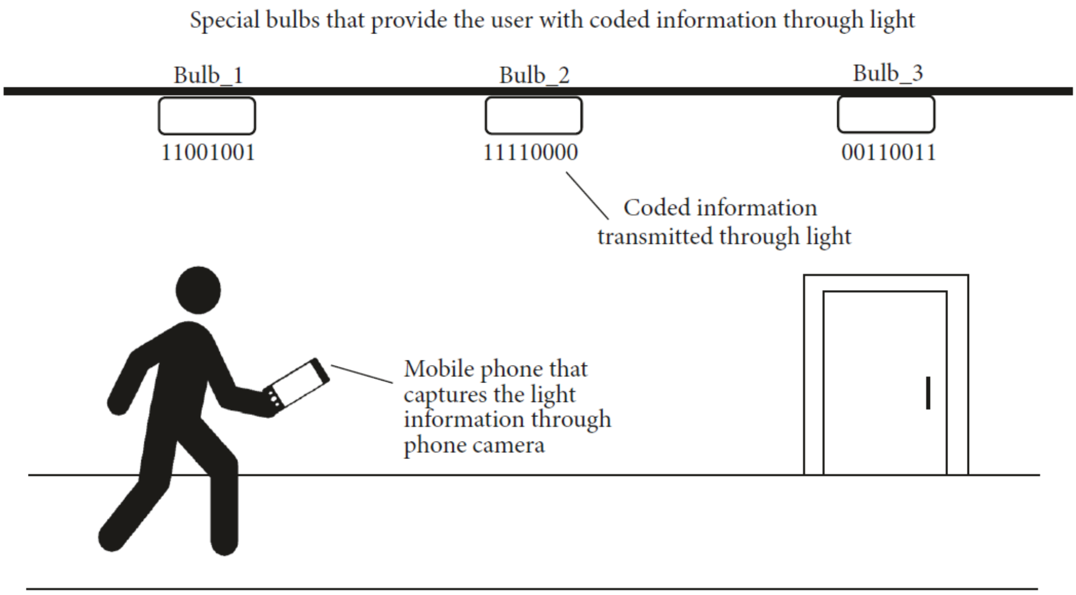


Figure 4.6 Visible Light Communication [24]

4.2.6 Radiofrequency

The use of radiofrequency technology in Indoor Positioning Systems (IPS) is common. This technology includes the widely known Wi-FI, BLE, RFID and UWB. These technologies are very promising for positioning in indoor environments, but accuracy and implementation costs need to be taken into account to select the most suitable among those for each application.

4.2.6.1 Wireless fidelity (Wi-Fi)

The IEEE 802.11 standard, widely known as Wi-Fi, was primarily used to allow Internet access to multiple devices either in private or commercial areas. Primarily, Wi-Fi range was about 100 meters while these days has increased to 1 kilometer in IEEE 802.11ah standard. A wide range of devices such as smartphones, personal computers, video-game consoles, tablets, laptops and many others which are used in the Internet of Things area allow Wi-Fi access. That fact makes Wi-Fi a suitable technology for indoor localization as it can be found almost everywhere. In large areas such as shopping malls and airports the already installed Wi-Fi access points of the roof on those areas might be useful as reference points to serve localization. That means that additional infrastructure may not be necessary in such areas for positioning. However, existing Wi-Fi networks were designed for improved communication between the devices and not localization, thus complex and efficient algorithms are needed to improve accuracy. Furthermore, accuracy depends on the number of access points and a small number of APs, as the system was designed primarily, may reduce accuracy.

The localization accuracy of techniques that use Wi-Fi technology is normally five to fifteen meters and depends on the complexity of the indoor environment, the number of people present in the area and the internal noise. Common techniques used with Wi-Fi are fingerprinting with RSSI or CSI, ToF and AoA or combination of those. Smartphone inertial sensors can also be used in combination with Wi-Fi to increase accuracy. RSSI and CSI measurements are used to construct the fingerprint database. RSSI was more used in applications since its collection is easily obtained from a commercial AP without the need for additional hardware. But the presence of fluctuations in RSSI values deteriorates the system's performance.

The Figure 4.7 illustrates the installed Wi-Fi access points in areas like a business hall, an office, a lounge and a hotel.

Figure 4.7 Illustration of Wi-Fi access point installed on roof in several areas

4.2.6.2 Bluetooth Low Energy (BLE)

The IEEE 802.15.1 standard, widely known as Bluetooth, is capable of supporting wireless communication between devices over short distances. The new Bluetooth version known as Bluetooth Low Energy (BLE) can be used to a range of seventy to a hundred meters while providing 24 Mbps. This technology is thought of as a competitor of Wi-Fi due to the wide availability of BLE to most mobile phones. Recently, BLE localization is used as iBeacons in smartphones. Beacons are small devices that can be installed on the walls [2]. A photo of a beacon is presented in Figure 4.8.That allows the localization of the smartphone in shopping malls, train stations, airports and other large areas where in-door localization may be implemented. A standard application for commercial usage of this technology is combined with providing shopping advice, navigation in a mall or unique offers that correspond to the user's specific needs. There have been proposed two BLE protocols, iBeacons, from Apple Inc. and Eddystone from Google Inc. Bluetooth signals are detected by beacons and then the location of the user is determined relative to beacon's location using the signal's strength. This technology could have promising re-sults in localization as most of the smartphones support Bluetooth and the last models also support BLE. Several studies recommend the usage of BLE technology or a combi-nation of Wi-Fi and BLE standards [25].

The main advantage of BLE technology is its power efficiency with power consumption of about 0.367mW which allows mobility along devices with a simple replacement of batteries. One bottleneck is that signal interference is usually present and may lead to inaccurate localization of the user and that complex signal processing is required to achieved good accuracy.

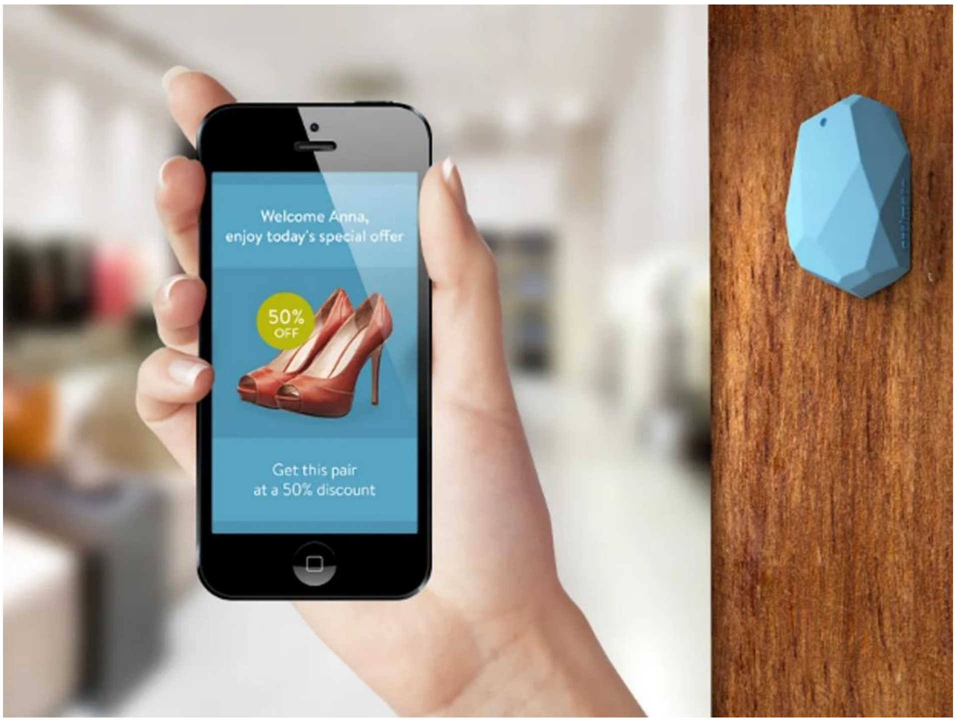


Figure 4.8 Beacon technology for offer promoting

4.2.6.3 ZigBee

ZigBee is built upon the IEEE 802.15.4 standard and is used in wireless networks for transfers across short distances between devices. Although it is inexpensive, it has a slow data transfer rate and is meant for low-power applications.

Primarily, ZigBee technology was intended to be used in applications such as smart home automation applications, healthcare and traffic light control, among others, due to its energy efficient operation and increased security. Such a localization system based on ZigBee technology includes a network of sensors.

Zigbee has also been used in indoor localization systems because it is cost efficient, power effective and allows easy access to RSSI values without the implementation of extra hardware. An indoor localization system based on ZigBee technology includes a network of sensors and algorithms that use the RSSI values to provide location estimations. The techniques used are the same as the ones used on Wi-Fi and BLE.

4.2.6.4 Radiofrequency Identification (RFID)

The term Radiofrequency Identification (RFID) describes a system that uses radio waves to transmit the identity of a person or object. It is a traditional identification technology which is mainly met in applications where objects need to be recognized in a large system. Medical equipment tracking, personnel or vehicle access control, logistics are some examples of applications.

The basic principle of RFID technology is to exchange radio signals in different frequencies between the RFID readers and the RFID tags. For example, if objects need to be tracked, RFID tags, which consist of a microchip and a radio antenna, are attached on them [26]. That RFID tag can emit radio signals which carry information, namely its unique ID. The RFID is scanned by the RFID reader, which consists of an antenna, processor, transceiver and an interface to allow connectivity to a server.

RFID tags can by classified as passive or active depending on how they get energy to send feedback to the reader. If they respond using the small amount of energy sent by the reader, they are called passive. If they have power supply on their own, then they are called active. There exist also semi-passive RFID tags which have their own battery but give a response only when a signal by an RFID reader is detected.

In terms of localization, these technologies have been used in application when the user or object was not needed to be tracked all along, but only when it passes through specific spots or control gates. In order for localization to take place, the system requires information about the signal intensity of each label reader to estimate the location of RFID tags. Some systems need to regularly scan the power levels to calculate the signal strength of the tag. An additional issue is that due to energy loss of the tag which works with battery, there is great fluctuation in RFID tag behavior.

4.2.6.5 Ultrawideband (UWB)

The technology of Ultrawideband (UWB) has been fundamentally used for communication systems of short range. Such systems could be found in indoor applications such as keyboard-Personal Computer connectivity. UWB also has industrial or medical applications such as radar transmissions, micro air vehicles and asset management. A main advantage of this technology is its durability against interference phenomena and fading as it occupies large bandwidth, short wavelength, fast communication, high data rate while conserving energy. Its different spectrum among other signals improves its durable properties against interference. Furthermore, a UWB signal can deal with complex indoor environments since it can pass through obstacles and walls, as it can occupy low carrier frequencies. However, there are some materials that can affect the UWB signals, such as liquids and metals.

In terms of indoor localization, UWB technology allows the detection of the main path in case of multipath phenomena which makes it more immune to multipath effects. That characteristic makes this type of signal suitable to use in localization techniques such as ToF and TDoA. UWB can achieve better accuracy in localization than BLE and Wi-Fi. Localization with UWB is a specialized solution which needs specially designed elements; thus, it is more suitable for applications in industrial environment.

4.3 Evaluation metrics of Indoor Positioning Technologies

4.3.1 Accuracy

The term accuracy in Indoor Positioning Systems refers to the average Euclidean distance between the estimated position of a target location by the system and the actual location. Accuracy is of high importance in such a system and the system has to deal with many challenges to obtain it. The complex indoor environments play a significant role in the system's accuracy because the system must overcome errors in localization due to multipath effects and existing obstacles in space. Additional environmental noise due to human presence is an extra challenge. The system may have to obtain noise reduction to deal with these noise issues while vast signal processing techniques may be required. Of course, increased accuracy is not the major thing in each application, but it depends on the special needs of each case. For example, some applications such as in-store offers may not have special needs in terms of accuracy but other metrics like cost or energy consumption may be more important.

4.3.2 Availability

Availability is extremely critical because the selected localization technology should be widely available to the user and not be restricted due to the existence of specialized hardware. The system should be easily adopted by users so careful selection is needed. UWB technology provides great results in terms of localization accuracy. However, this technology is not widely available as a small number of devices support this and would not be a smart choice. On the other hand, Wi-Fi is present in most of the devices and for that reason systems that use Wi-Fi technology are much easier implemented and widely used.

4.3.3 Security

Previously, security was not considered in localization technologies. However lately, due to personal data protection, much concern should be given to that field. If we consider localization and tracking of the users in a shopping mall where the consumer behaviors are being recorded, we can detect the importance of privacy and security of the system's users. The personal information of the users may be vulnerable to misuse from unauthorized access because businesses are extremely interested in the way that clients behave and interact with offers or recommendations in stores. Privacy and security mechanisms regarding localization are not only restricted to customers in malls but must be applied in every application where personal information is collected.

4.3.4 Scalability

Scalability ensures that the system's operation remains unaffected or slightly affected when it is used in a larger application. Profoundly, a system that achieved great accuracy in a small area may behave poorly in a larger space. Comparable results should be obtained either if the system is used in a small room or in a huge space like an airport. It depends also on the distance between the receiver and the transmitter. It also should support and provide localization services efficiently to all users independently of the number of users connected.

4.3.5 Cost

The cost of a positioning system may be translated to several factors like the funds needed, the time required, the space that the system occupies and the energy it consumes. These costs do not refer only to the system's construction process, but the maintenance time and costs should be considered. An efficient system should require as little additional infrastructure as possible. That would lead to the wide and easy adoption of the system not only by businesses that can afford the implementation of such a system but also from companies that can implement that using the existing infrastructure. The cost of the system is the feature that directly affects its popularity.

4.3.6 Energy Consumption

The energy efficiency of a localization system should seriously be taken into account as its main purpose is to serve the users and not hinder them. For example, when someone is using an indoor localization service for navigation in a shopping mall, the operation of the system should not have unwanted results such as draining the battery of the user's smartphone. That will result in the user's unsatisfaction and the attitude of the users against this technology will be negative. The frequency and the transmission power of the signal affect the power consumption. The higher these values, the higher the power consumption. Another factor which demands higher amounts of energy is the computational complexity of the localization process. The Figure 4.9 compares several technologies used in localization in terms of cost, accuracy and energy consumption.

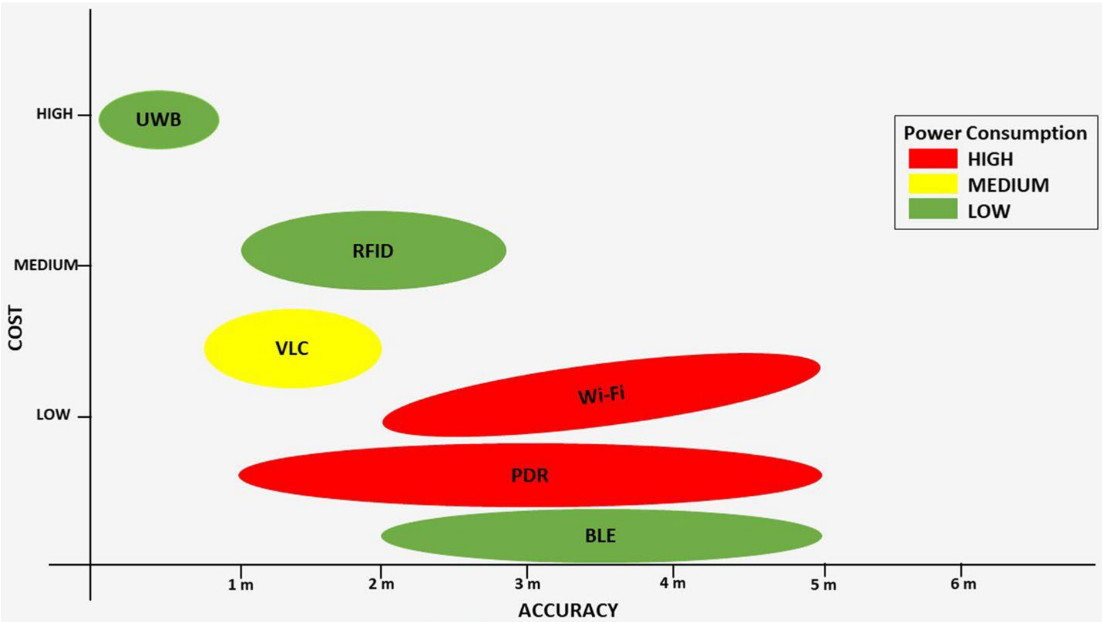


Figure 4.9 A comparison of indoor localization technologies [27]

4.3.7 Coverage space

The area that the localization system can cover efficiently is very important because it can directly affect the cost of the entire system. The larger the area a system can cover efficiently, the smaller number of reference nodes needed to estimate the location of the user.

4.3.8 Real time

Accurate real-time coverage is an important aspect in some indoor localization applications. Such applications could be, for example, a safety system which tracks and navigates people in emergency situations. Extensive signal processing and noise reduction techniques used in localization algorithms may introduce some delay in real-time positioning, since the more complex the system is, the more time-consuming is turned into. Indoor localization services should be designed to provide as much negligible delay as possible when important levels of accuracy are needed, while this also increases the system's costs.

5 Neural Networks

This chapter contains principles of Neural Networks. It will present how a Neural Network works but also will explain fundamental notions and algorithms such as Gradient Descent and Backpropagation. The working principle of Convolutional and Recurrent Neural Networks will also be explained.

5.1 Structure of a Neural Network

To better understand the way that the Neural Network was founded, the examination of the neuron is important. The neuron, which imitates the way the human brain works, is the basic component of all Neural Networks. In the Figure 5.1 the communication between two neurons is shown. The connection is achieved with those tentacles which exist perimetrically the nucleus of each neuron and are called dendrites and also the tail or branching fibre which is called axon. The perception of their surroundings a human can understand is formulated with electrical signals that flow through the dendrites and axon.

An interesting thing is that the axon is not in touch with the dendrites but there is a gap between them and the transfer of the signal is obtained through a process called synapse. Each neuron is connected to many thousands of other neurons and as a result receives constantly an enormous number of incoming signals [28]. These signals reach the cell body where they are summed together and if a specific threshold is exceeded that will cause the neuron to fire up. That impulse is then transmitted to other neurons via the axon.

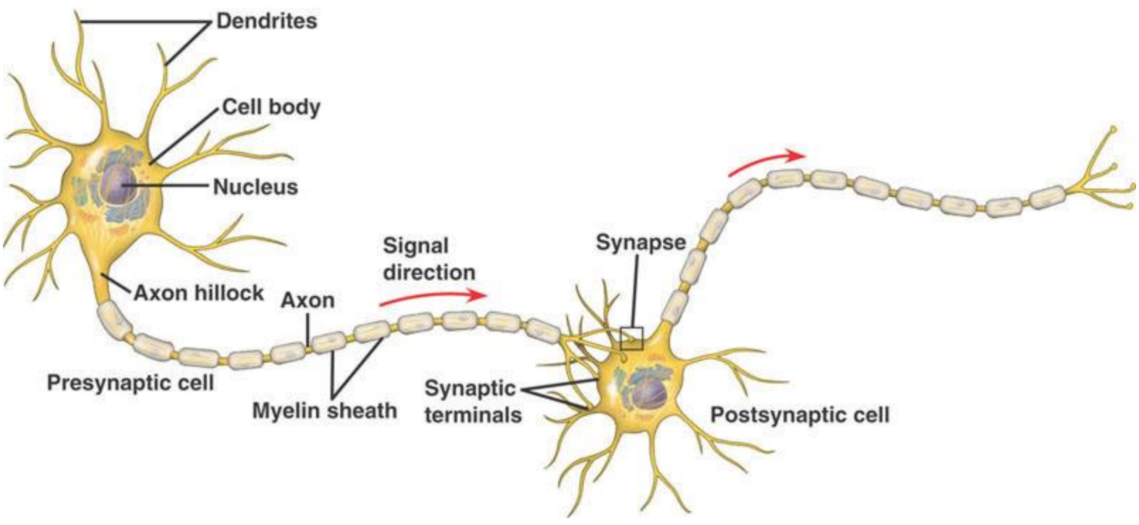


Figure 5.1 Communication between two neurons [29]

The Figure 5.2 shows a model of an artificial neuron. The input signals on the left represent the signals that enter the neuron which is known in the middle of the figure. These incoming signals in a human brain neuron could represent touch or sight. Similarly, in a Neural Network these inputs are the independent variables. They enter the main neuron and then exit the other side as output values. But something that one must be aware of is that these input values should not enter the network in their raw form, but a preprocessing step is always a good approach. Therefore, they must be either normalized or standardized. That part is crucial for the network because it transforms the input variables in a similar range and makes the process by the network less complex.

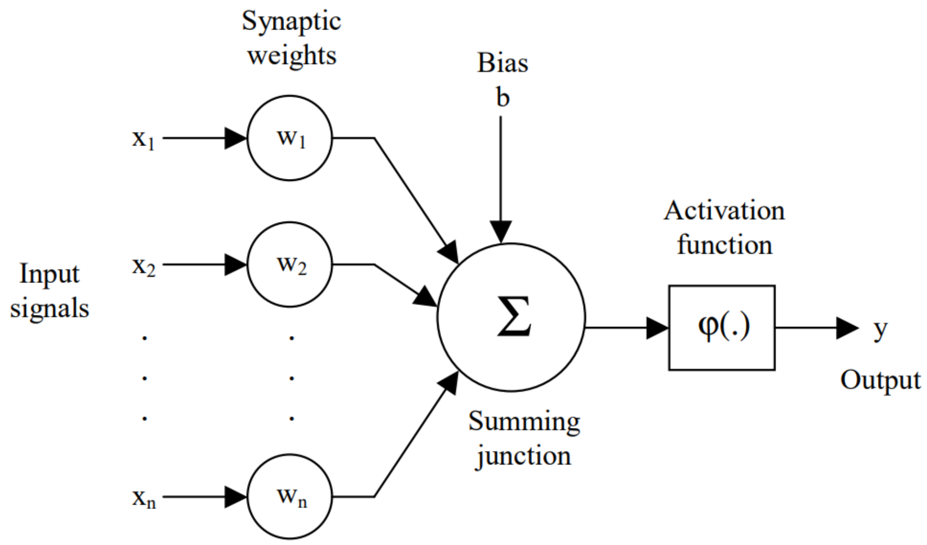


Figure 5.2 Model of an artificial neuron [30]

In mathematical terms, a neuron may be described by:

$$y = \varphi(\sum_{j=1}^{n} w_j x_j + b)$$

where $x_1, \ldots x_n$ are the incoming signals shown on the left part of the Figure 5.2, $w_1, \ldots w_n$ are the weights of the synapse, b is the bias, φ is the activation function and finally y is the output signal which is shown on the right side of the neuron.

A weight is assigned to each synapse. The weights are the basis of the learning process of the Neural Networks, because based on those the Network is able to decide what information must be taken into account and what must not. More precisely, the weights decide which of the incoming signals should pass along or at what level that should be done. Through the training process of a Neural Network the weights are adjusted in order to obtain the best result at the output.

Next, the alteration that undergo the incoming signals will be discussed. The first action to be taken by the neuron is the summation of the weights. Then, an activation function is applied either just to the neuron or a layer of neurons which is what decides if a signal passes along or not as discussed previously. There will be more discussion on the activation functions in the next subsection.

A combination of the input variables that enter the network are not stand alone but all together constitute a single observation. For example, a single observation may be the humidity, wind velocity and ambient temperature at a specific location. Many similar groups make up many observations that are going to feed the neural network. It is essential to make it clear that both the input values and also the output constitute a single observation. The output value is the so-called ground truth which together with the inputs will be used to train the Neural Network.

On the other side the output values can take several forms. They can be continuous, when for example we predict a temperature at a location given other ambient characteristics, they can also be categorical and finally binary which take for example values yes or no. A classic example of a binary outcome is the prediction if a customer of a given company will churn or not.

5.2 Activation functions

The activation function is the process that is applied to the weighted incoming signal after its entrance in the neuron. This function takes those inputs and produces a new value which is nothing more than the output. For the neuron to produce that output, an activation function is required. There are many different activation function types, which means there are numerous methods for handling the input values. Each of those functions has its own specific properties and should be applied to a specific application where it is suitable. As mentioned previously, the output of the preceding layer plus the bias are weighted together and sent to the activation function. The bias directs the activation function in a manner that causes the model to fit our dataset. The activation functions can be divided into two types, namely linear and non-linear activation functions. Next, several types of activation functions are examined.

5.2.1 Linear Activation Function

In the case of the linear activation function which is shown in Figure 5.3 There is a proportional relationship between the weighted sum and the output. This function can handle multiclass problems, but a huge disadvantage is that independently of the depth of the network, the last layer of neurons will be a function of the first layer which results in the restriction of handling and identifying complex patterns by the network.

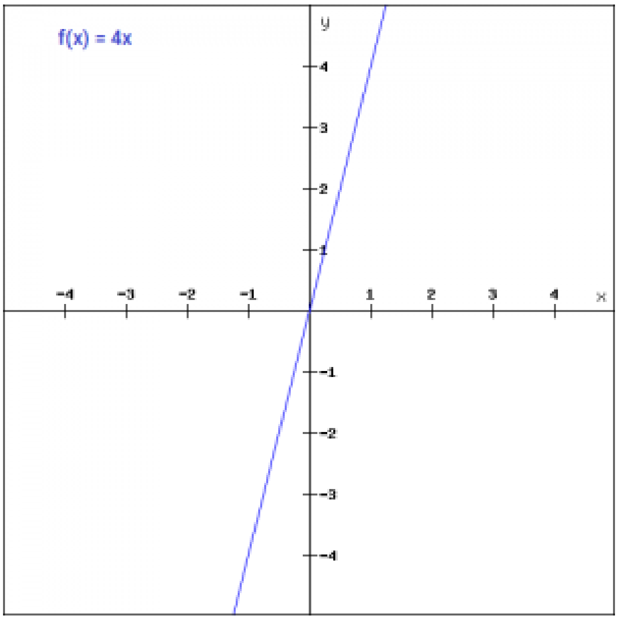


Figure 5.3 Linear activation function [31]

The mathematical expression of the linear activation function is:

$$f(x) = ax$$

where a is a constant value chosen by the user.

The gradient is equal to the constant that has been chosen and is the same for each iteration during the training process which leads to incapability of improving the error between the predicted and the actual value.

5.2.2 Non-Linear Activation Functions

In this section some of the non-linear activation functions used in Neural Networks will be discussed.

5.2.2.1 Threshold or Binary activation function

That is a simple type of activation function. The Figure 5.4 shows what a threshold activation function looks like. The weighted sum of the incoming signals is represented in the x-axis. The y-axis contains the values from 0 to 1. The way that it works is the following. If the weighted sum on the x-axis is less than 0, then the activation function, here the threshold activation function will pass along the value 0, otherwise it passes along the value 1. In other words, after a certain threshold, the neuron is activated and if below that threshold that leads to the deactivation of the neuron. That type of function can be used in binary outcomes, taking also in mind the name of the function, when for example we have to deal with a yes or no classification. However, this function cannot be used when we have to handle more than two classes. Also, the gradient of that function is zero. That fact may hinder the process of back propagation. That happens because the calculation of the derivative of $f(x)$ will be equal to zero. Back propagation will be discussed in the next sections.

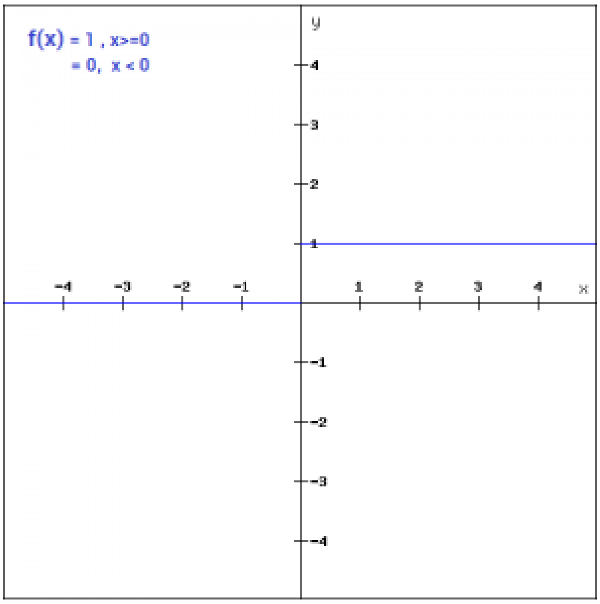


Figure 5.4 Theshold or Binary activation function [31]

Mathematically the threshold or binary activation function can be defined as follows:

$$f(x) = 1, x \geq 0$$

$$f(x) = 0, x < 0$$

5.2.3 The Sigmoid or logistic activation function

The fact that it is a non-linear function leads the sigmoid activation function to its wide use. The sigmoid function is shown in Figure 5.5 and its graph is known for its s-shaped form. This function is used in models where the output is the prediction of a probability since the probability of an event lies between 0 and 1. But this function obtains almost no change in the prediction of very high or very low inputs which leads to a problem known as vanishing gradient that hinders the neural network from being trained further. Also is continuously differentiable and appears commonly in Statistics as a cumulative distribution function. It is sometimes desirable to have the activation function in the range from -1 to 1 while that form of the sigmoid function is defined as the hyperbolic tangent function which will be discussed in a next subsection.

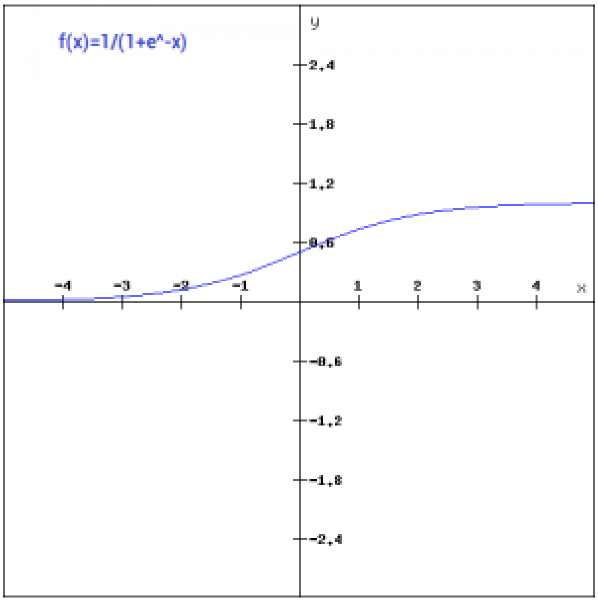


Figure 5.5 The Sigmoid or Logistic activation function [31]

The sigmoid function can be mathematically expressed by the formula:

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

5.2.4 The Tanh or Hyperbolic tangent activation function

The hyperbolic tangent function which is shown in Figure 5.6 is similar to sigmoid function but it has better properties since it allows negative outputs. Its range is from -1 to 1 and is also s-shaped. The negative inputs will be mapped strongly negative and also the inputs near 0 will be mapped close to 0 too. The hyperbolic tangent function and also the sigmoid one is widely used in feed-forward neural networks. However, tanh suffers also from the vanishing gradient problem as happens with the sigmoid. It is preferred over sigmoid since its gradients are not obligated to vary in a certain direction.

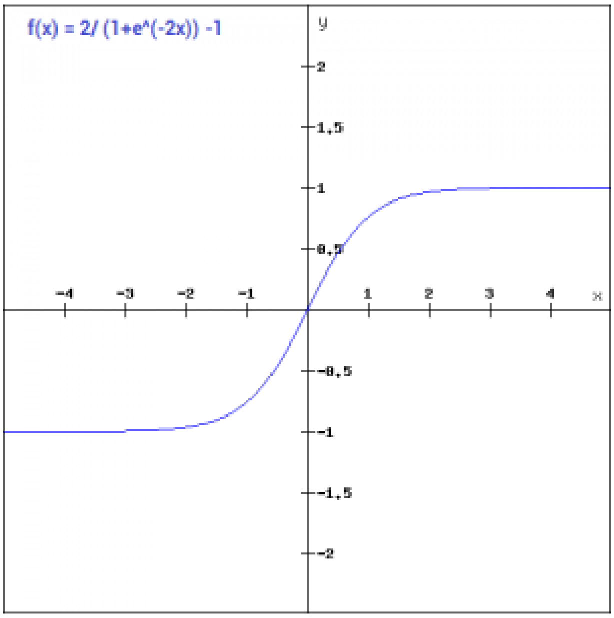


Figure 5.6 Tanh or Hyperbolic Tangent activation function [31]

The mathematical representation of the hyperbolic tangent function is:

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

5.2.5 The Rectifier activation function

This function is one of the most popular activation functions used in Neural Networks and is shown in Figure 5.7. It is called ReLU which stands for Rectified Linear Unit and is also a non-linear function. Even though it is simple, the constant gradient of that function leads to a faster learning rate. The advantage of ReLU is that the activation of all the neurons does not happen simultaneously, but only a specific number of neurons are activated at the same time. A neuron's deactivation will occur only when the linear transformation's output is 0.

It happens sometimes that the gradient becomes zero and as a result the update of the weights and biases does happen during back-propagation which leads to poor network training.

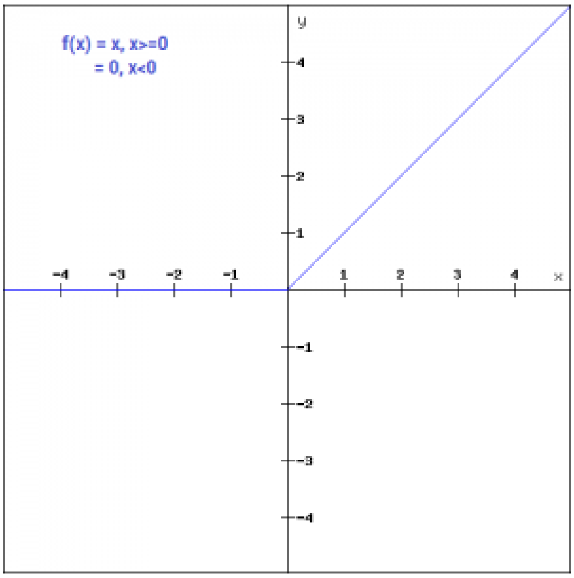


Figure 5.7 The Rectified linear unit (ReLU) activation function [31]

The mathematical definition of the Rectified linear unit activation function is:

$$f(x) = \max(0, x)$$

In the following Figure 5.8 a summary of several activation functions is presented.

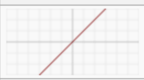
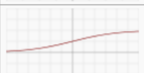
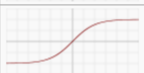
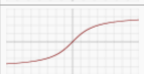
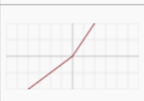
Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1-f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1+e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Figure 5.8 Summary of several activation functions [32]

5.3 Neural Network

If we consider the Figure 5.2 where a basic Neural Network is presented, we can see that on the left side there is a single row of input variables, or one single observation as it was mentioned. The weighted synapses are shown in circles which enter the single neuron in the middle of the figure. On the right side there is an output which is defined as y. This graph represents a single-layer Feed-forward Neural Network.

In the next Figure 5.9 we can see that the weighted inputs are summed up, the bias is added, the result passes along an activation function $\sigma(y)$ and finally the output y is obtained. But here we should focus on the fact that the output y has been replaced by $\hat{y}$. The core of the learning process is the difference between the two values y and $\hat{y}$. The $\hat{y}$ is the predicted value which was the output of the network while y is the actual value which belongs to the observation, that single row in the dataset. The comparison of those values from the network is the way that the network learns. In that process the help of a cost function is needed, which will be explained in the next section.

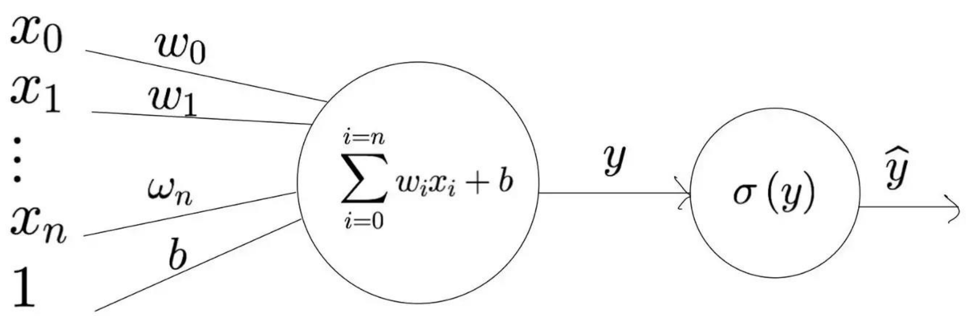


Figure 5.9 Graphical representation of a perceptron [32]

5.4 Cost function

Training a neural network means optimizing the network's parameters, which are the weights and biases. In supervised learning applications like regression and classification, the goal is to minimize the error between the network's predictions and the actual values of the observations [33]. More precisely, that error is used to form a cost function, the minimization of which through training is the target of the entire process. For regression tasks, a cost function widely used is the *mean squared error* and is expressed as follows:

$$J = \frac{1}{N}\sum_{i}^{N}\left(y^{(i)} - \hat{y}^{(i)}\right)^2$$

where N is the total number of observations, $y^{(i)}$ is the i^{th} actual value and $\hat{y}^{(i)}$ is the predicted value. For classification applications another type of cost function is used and is called *cross entropy* cost function, but actually there we called it a loss function and not cost function anymore. The mathematical expression of cross entropy is the following:

$$J = -\sum_{i}^{N}\left(y^{(i)} log\hat{y}^{(i)}\right)$$

where N is the total number of observations, $y^{(i)}$ is the i^{th} actual value and $\hat{y}^{(i)}$ is the predicted value.

After finding the suitable cost function depending on the application the trained parameters, namely the weights and biases, of the network are updated iteratively with optimizer algorithms. Optimization algorithms play a key role in Machine Learning. Depending on the optimizer we select, the model will operate in a different way. The most popular of those algorithms are the Adam and the Stochastic Gradient Descent. That iterating process continues as long as there is a strong difference between the actual and the predicted value [34]. The resulting data are fed back to the entire network, the weights are adjusted and a new cost function occurs. This repetitive process is completed when the cost function gets as small a value as possible. The smaller the cost function the smaller the error in the predictions and so higher accuracy is achieved. If the predicted values are close as possible to the ground truth that means that the weights are optimized and the training process is done.

Independently of the number of observations there is in the dataset, the comparison between the ground truth and the predicted values will take part and the cost function will be applied to all of the observations together. Then, the adjusting of the weights will occur.

5.5 Gradient Descent, mini-batch and stochastic gradient descent

As mentioned previously, the cost function is the difference between the predicted value, which is the output of the neural network, and the ground truth. The model's accuracy is increased as the cost function's value decreases, meaning the output values of the model get closer to the actual values.

The adjusting of the weights occurs with a process called backpropagation when the end data are fed back through the artificial neural network. This happens iteratively and hopefully the value of the cost function is decreased. The value of the cost function is directly affected by the weights. Since the output values of the model are determined by the weights, the more optimized those are the less will be the cost function. One could suppose that easily all the possible combinations for the values of the weights can be tested and, in that way, the corresponding values of the cost function can be calculated. The specific weights that give the lowest value for the cost function are the optimal. But as the network's size increases, that attempt is increasingly unachievable, impractical and computationally costly. If we take a look at the deep artificial neural network in the Figure 5.10, which was used in [35] to predict trends in dissolved oxygen, there is an enormous number of combinations for the values of the weights. Finding the lowest value of the cost function of such a network is a time-consuming operation. In the case of a deeper network with more layers than that in the Figure 5.10 the time demanding operation would also require even greater hardware resources.

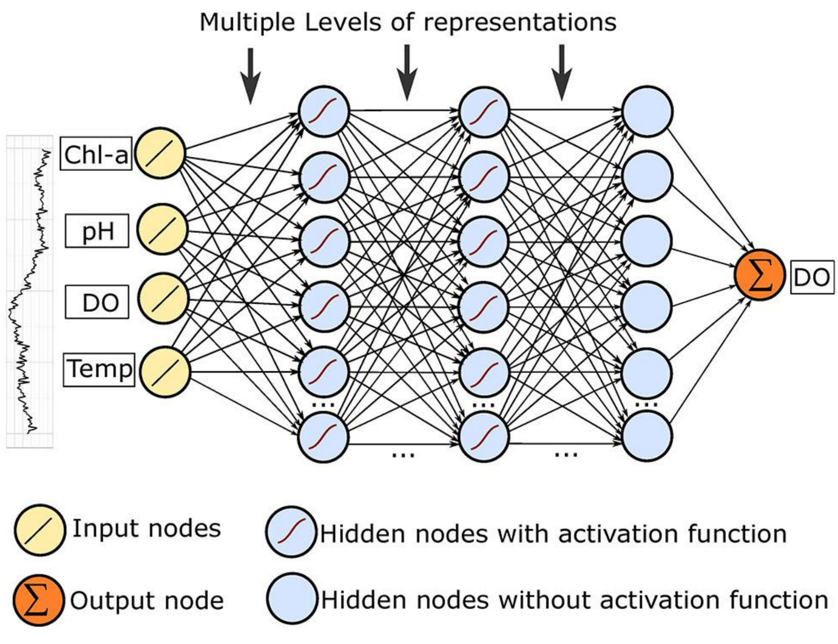


Figure 5.10 Multilayer Feed Forward Neural Network [35]

In some cases, we may have an almost infinite number of weights to optimize. That problem is called the *curse of dimensionality*.

The gradient descent algorithm provides a solution to such issue. The objective of the optimization algorithm gradient descent is to minimize the cost function. Generally, this function is also referred to as objective function but in machine learning terminology it is called cost or loss function.

As gradient of a function at any point we mean the direction of ascent of the function at that specific point. In the case that we want to maximize a function we just have to pick a starting point at random [36]. Then we have to compute at that point the gradient and take a small step in the direction which causes the increase of the function. After that we repeat that starting from the new point we ended on the previous step. In minimizing a function, we follow the same steps but in the opposite direction, which causes the function to be minimized.

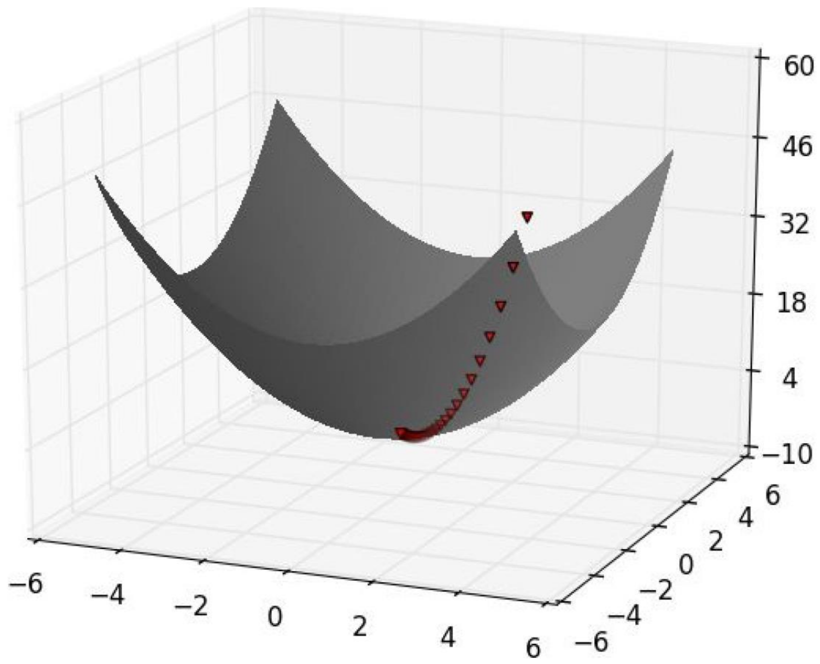


Figure 5.11 Finding a minimum of a function using gradient descent [36]

In summary, the following are the steps involved in utilizing gradient descent to find a function's minimum. First, we take the gradient by calculating the derivative of the function with respect to the certain parameter. If there are several parameters to consider, we take the partial derivatives with regard to each parameter. We then have to multiply the learning rate with the derivatives of the different parameters and -1.

Graphically this can by illustrated in the Figure 5.12 where we see that the learning rate we select is crucial for finding efficiently the minimum. Simply, After the calculation of the slope of the function at the starting point we direct ourselves downhill to the minimum values. At each iteration, namely at each step, the weights and the biases are adjusted. In the following equation the collection of the weights is notated as w. For simplification, the notation of biased is not considered.

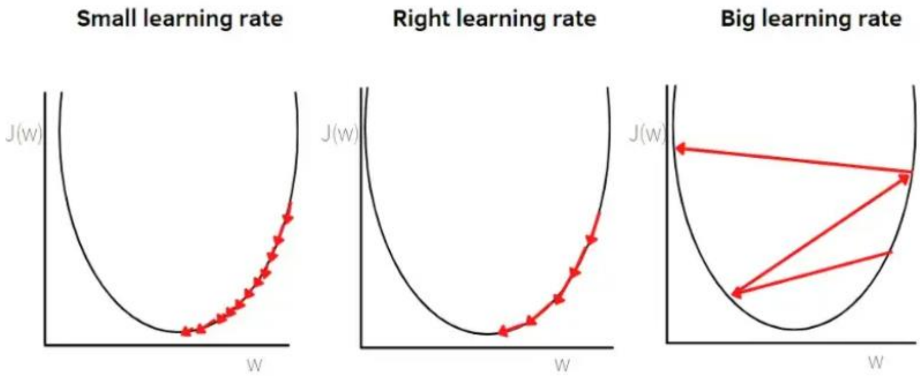


Figure 5.12 Gradient descent with different learning rates [32]

The expression of the gradient update is:

$$w_{k+1} = w_k - \eta \nabla J(w_k)$$

Where w_{k+1} is the collection of the weights at the k+1 step of the algorithm, w_k is the collection of the weights at the k step, η is the learning rate or step size and $\nabla J(w_k)$ is the gradient of the cost function at step k.

After the update of the weights, the calculation of the gradient is repeated for the new weights collection. As it is clearly shown in the Figure 5.12 the step size is critical for the minimization of the function, as a selection of a big value for that may lead to the divergence of the cost function. Similarly, if we select a small value for the learning rate it will converge but it will be too time-consuming. So, the appropriate value should be found. In that method we need to take the gradients for the whole dataset in order to make a single update

In the gradient descent method presented, which is called *batch gradient descent*, the observations of the dataset are evaluated at once. We need to take the gradients for the whole dataset in order to obtain a single update. That will force the algorithm to be terribly slow and set it intractable for big datasets, which drain the memory [37]. This method is preferred when the dataset is relatively small. For bigger datasets *mini batch gradient* descent is another technique where the data are evaluated in batches. The entire dataset is split into a number of mini-batches and the gradient descent is applied to each of those. The mini batch gradient descent is much less computationally costly than the batch gradient descent. In the Figure 5.13 the difference between batch and mini-batch gradient descent can be observed.

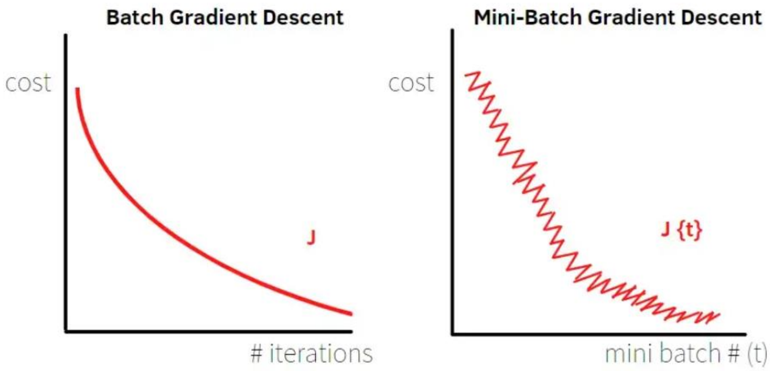


Figure 5.13 Comparison of batch and mini-batch gradient descent [32]

In the case of batch gradient descent, the cost function is decreased at each iteration since the whole dataset is evaluated. In the case of mini-batch gradient descent, a mini-batch is evaluated at each iteration and as a result, there may be batches which do not lead to the minimization of the function as they may contain some noisy data. That leads to that zig-zagging behavior shown in the Figure 5.13. In cases when there are noisy observations the use of too small mini batches may lead the algorithm not to converge.

The cost function that was presented in Figure 5.12 was convex and the global minimum of the function could be easily found by gradient descent. In real applications one can end up with a cost function like the one presented in the Figure 5.14. When we choose a cost function that is not the square difference between the actual and the predicted values, we get a non-convex cost function. In that case, where the cost function is not convex, there is one global minimum but more than one local minimums. The gradient descent method may get stuck on a local minimum as it is shown in Figure 5.15 and never reach the optimal weights which minimize the cost function. *Stochastic gradient descent* is an alternative method which provides a solution to that problem.

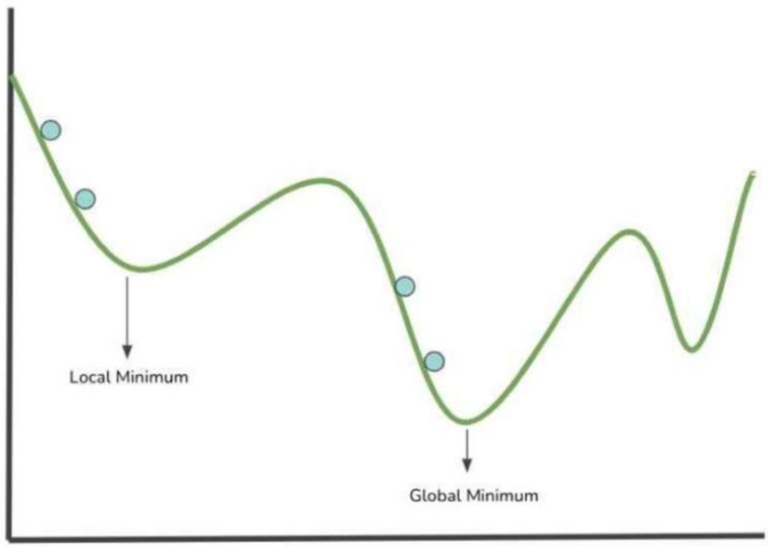


Figure 5.14 Local and global minimums of a cost function [38]

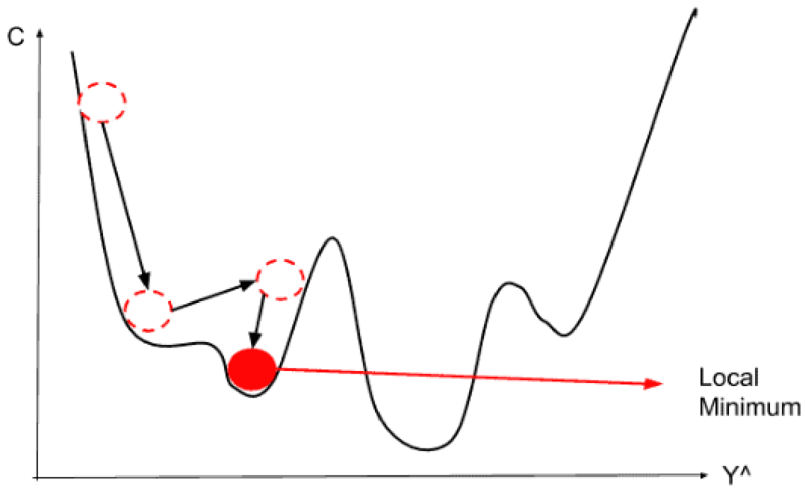


Figure 5.15 Inefficient minimization of a cost function [39]

Gradient descent and stochastic gradient descent differ in how these methods update the weights. In gradient descent all the observations in our dataset are fed into the neural network simultaneously and the weights are adjusted. Instead, in the stochastic gradient descent method an observation is inserted into the neural network and the weights are adjusted based on the cost function. Then, we move to the next observation, calculate the cost function and the weights are adjusted. We do so for the complete set of observations. In that way a kind of random noise is added which helps to get away from the local minimum and converge to the global maximum of the cost function. When updating the weights after calculating the cost function for each observation the fluctuations are higher and that helps to avoid the problem of stucking in a local minimum point.

5.6 Backpropagation

Backpropagation is an advanced algorithm which allows adjusting the weights of the Neural Network. It is used in layered feed-forward Artificial Neural Networks. That is, the neurons of the layers are forwarding their signals, then the errors are calculated and are propagated backwards. The backpropagation algorithm is used in supervised learning applications where we try to fit the model to the observations. The key role of this algorithm is the decrease of the difference between the actual values of the observations and the predictions of the model. That iterative process continues until the model fits the training data [40]. That means that the components that influence the output of a neuron, namely weights, biases and activation functions, are adjusted to reduce the error. The learning rate of the Neural Networks determines how much the weights are adjusted. The calculation of the partial derivatives of the cost function with respect to the weights, biases and the previous layer's activation functions, is the key mathematical process behind backpropagation [41]. The role is to identify the values that affect the cost function's gradient. By the terminations previous layer, in back propagation it is meant the layers closer to the output, since the process is happening backwards.

5.7 Convolutional and Recurrent Neural Networks

5.7.1 Convolutional Neural Network

Convolutional Neural Networks (CNNs) is a category of Artificial Neural Networks that try to imitate the way the human brain recognizes whatever is in our surroundings. The key role of those networks is to try to mimic the functionality of the human brain's visual cortex and detect objects visually. Briefly, the human brain categorizes the objects it sees depending on features it detects in them. Over the past years, these artificial networks have attracted much interest due to their high-accuracy performance in image classification applications.

The basic components that take part in the entire process of categorizing an image are the input image, the CNN and the output result. The output result is the class in which input image belongs to. However, that image classification is not the only use of CNNs, but they can also be used to detect human emotions of an image based on facial expressions.

The basis of the whole process is how the CNNs scan the input images. Black and white images are two-dimensional, while colored images are three-dimensional. Each pixel of a black and white image is assigned to a number between 0 and 255. It should be mentioned that an image's colors are represented on a scale from 0 to 255. In a colored image each pixel is represented by a separate value for each one of the three channels, namely red, green and blue, since any color of a pixel is a combination of the values of those three layers. The Figure 5.16 illustrates an example of the red, green and blue layers. Each layer of the matrix is a two-dimensional matrix of red, green and blue pixel values. A single color is represented in each one of the three layers with a number between 0 and 255 and the combination of those three values is the color itself. That combination represents the color to the neural network.

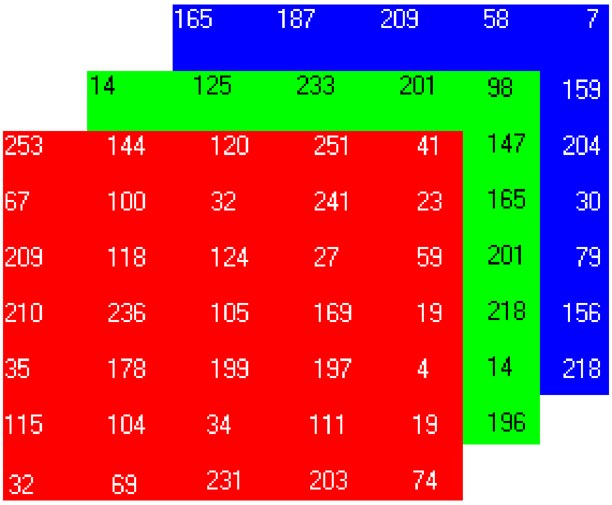


Figure 5.16 A three-dimensional RGB matrix [42]

Convolution

Convolution is the first step of a CNN operation. In the Figure 5.17 there are shown the three components that take part in the convolution process. These are the input image, the feature detector and the feature map. The feature detector is also called filter or kernel.

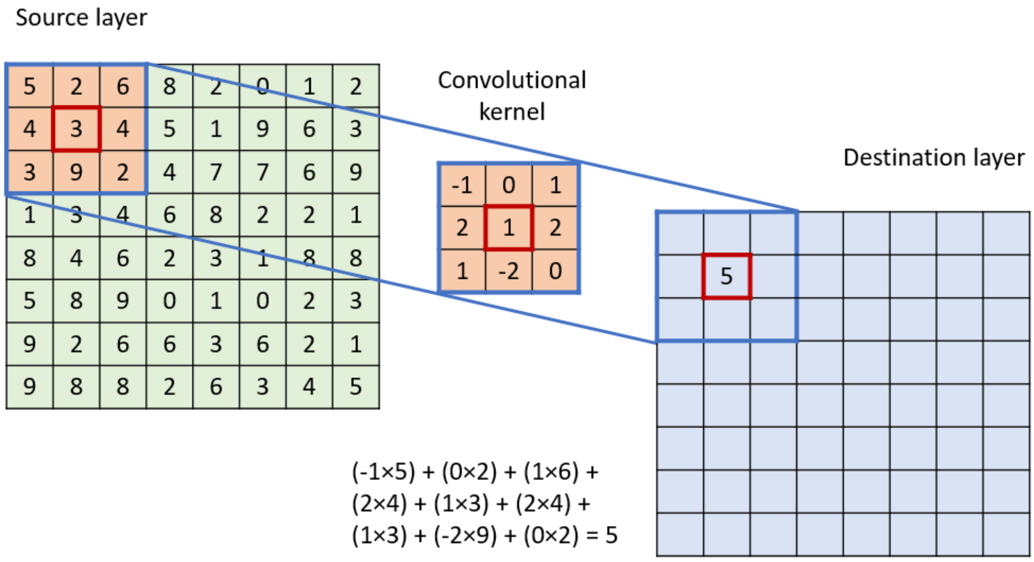


Figure 5.17 Convolution operation [43]

The most widely used size for the convolutional layer is a 3x3 matrix, but 5x5 or 7x7 matrices are also used. The way that the process works is the following. First, the convolutional kernel is placed on the source layer, namely the input image. It starts scanning the image from the top left corner as it is shown on the Figure 5.17. Next, it is the turn of the computation of the product between the number that are in the exact same location in the filter and the region that is demarcated in the input image. A single number is produced after summing up all the products between the two matrices, in the way it is shown on the figure above. Then the filter is moved to the right and the same process is repeated, namely the calculation of the new product etc. How much the filter is moved, it is called a stride. As a result, if the filter is moved by one pixel, that is referred as a stride of one pixel, if moved by two pixels then is referred as a stride of two etc. The resulting matrix, destination layers as shown above, is also called as feature or activation map.

One of the key roles of the convolution operation is the reduction of the input's size which is directly dependent on the value of the stride. It is profound that the size of the output image will always be less than the size of the original image and so some information is lost. But that is the role of the kernel, to detect the most prominent features and exclude the meaningless ones. That makes the process less computational costly and less time-consuming. One can also use convolution operation to sharpen or blur images.

In practice, during training a CNN, there are produced multiple feature detectors, so multiple feature maps are developed. These are known as convolutional layers. Several feature detectors are produced based on the features that the CNN selects features that are important during training.

ReLU

The Rectified Linear Unit is an additional step to the convolution operation. As already known, ReLU is a non-linear operation and it is used to increase the non-linearity in the input images. An original image contains many features that are characterized by non-linearity. After the convolution operation some linearity may appear in the output image. The ReLU is applied to bring back some non-linearity after every convolution operation.

Pooling

The role of pooling is to get rid of unnecessary information and recognize the features of the images that belong in the same class despite the presence of many differences among the images. In that way, the model will not be affected by the different colors, textures, the angles that the images were taken etc. Pooling is applied on the feature maps that have been produced after the convolution and the ReLU steps. It actually reduces the size of each one of the feature maps while conserving the most dominant features and disposing the noise in the images.

There are multiple types of pooling, but the most widely used are the max pooling, sum pooling and average or mean pooling.

The Figure 5.18 illustrates how the max pooling operation works. On the left part of the figure the is a rectified feature map which is actually the result after applying convolution and the ReLU step on an input image. In practice there are lots of these rectified feature maps in the CNN operation. A box of, regularly, a size 2x2 pixel is placed on the top left corner of the convolved and rectified feature map. In the case of max pooling, which we describe, we extract the maximum value of the ones that are in the box and we start creating a new matrix which is the pooled feature map. We move along the row, do the same and next we move down. In the specific example a stride of 2 was used, which is the most used in applications.

In summary, one benefit of pooling or down-sampling as it is also called, is the fact that it reduces overfitting since irrelevant information is lost. The size of the images is also reduced, and so do the number of parameters that are going to be fed in the layer of the feedforward NN after flattening.

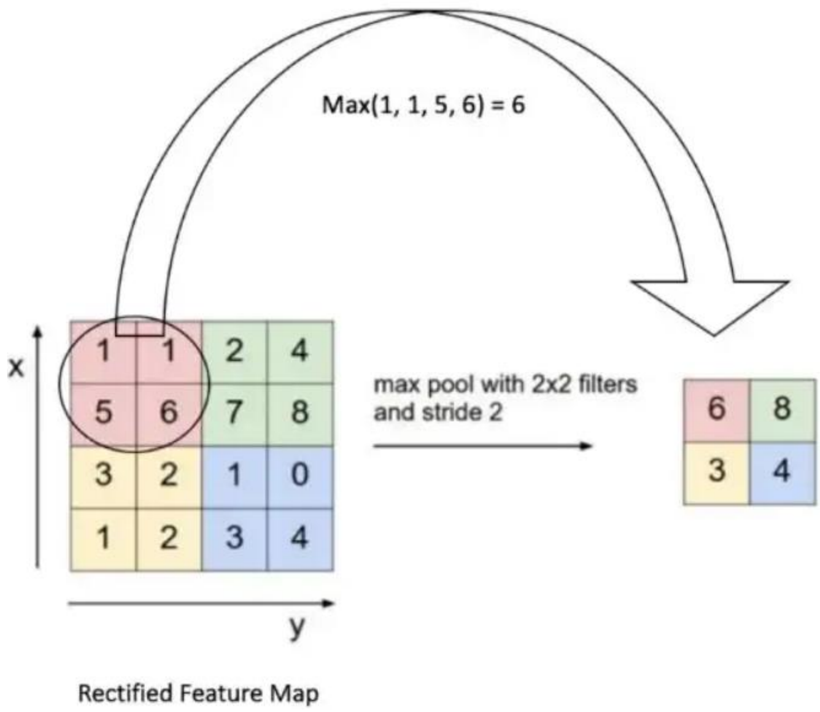


Figure 5.18 Pooling operation

Flattening

In this step, the pooled feature maps constructed in the pooling step are converted into a vector. As the name of that process indicates, the feature maps are flattened into a single column. The reason for this is that the flattened column will be fed into a feedforward artificial neural network. The following Figure 5.19 illustrates how one of the pooled feature maps is turned into a long vector after flattening.

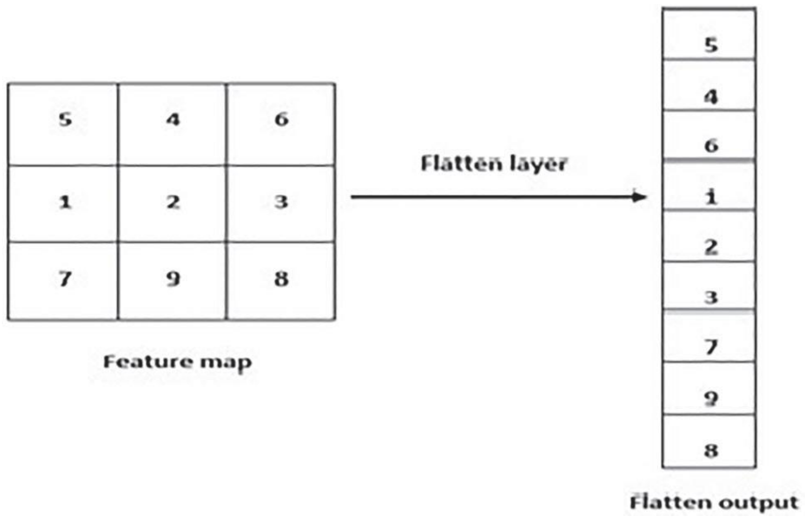


Figure 5.19 Flattening operation [44]

Full Connection

The flattened vector was created to be fed into a fully connected feedforward ANN. The term fully connected is used to express that all the neurons of the feedforward network in a CNN are connected to each other.

The flattened layer, which is the result after the convolution and pooling layers, contains all the dominant features of the images used from the feedforward network to classify the image.

Once a prediction is made, it is compared with the actual class of the image. In the ANN section the function that was used to calculate the error was called cost function while in CNNs that function is called loss function. After the loss function is calculated, similarly with ANN, the model should be optimized to increase the accuracy of the prediction. So, the error flows backwards and the weights of the ANN are adjusted but also the kernels at the convolution process alter in order to be able to detect new, more relevant, features of the images. That is an iterative process which continues until the network is optimized.

The output of the ANN are actually probabilities which indicate in what class the network thinks the image belongs to. These probabilities in their raw form are values which do not add up to one. For the summation of the probabilities for all the classes to be equal to one, the probabilities must by processed with the softmax function. The formula for the soft-max function is given by:

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

where $\vec{z}$ is the input vector to the softmax function, z_i is the value of each component of the input vector to the softmax function, e^{z_i} is the standard exponential function applied to each component of the input of the input vector $\vec{z}$ and K is the number of the classes in the classifier.

After the softmax function, the cross-entropy loss function is applied which is used in the same way as the mean squared error lost function in ANNs. One of the advantages of using the cross-entropy function is that it leads the network to be optimized more efficiently through the calculation of the error. This applies to classification problems. For regression ones, the mean squared error is preferred.

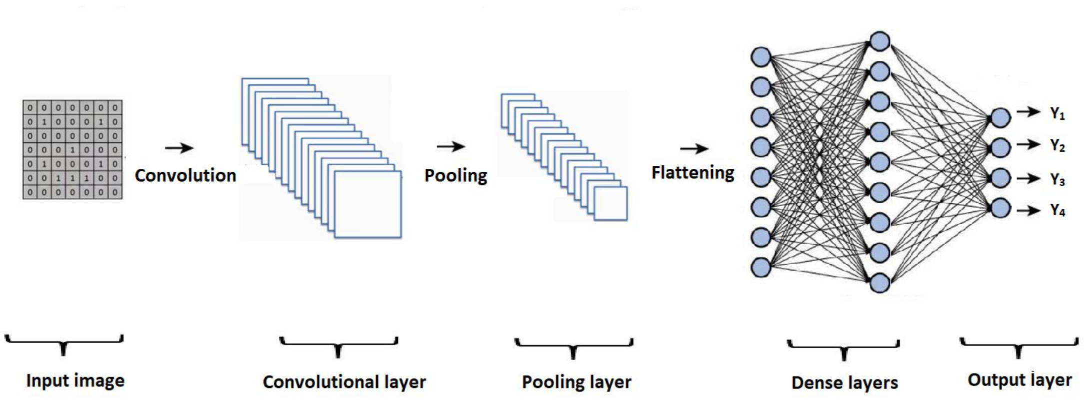


Figure 5.20 Typical architecture of a Convolutional Neural Network [45]

On the Figure 5.20 the typical architecture of a CNN is illustrated. Briefly the input image is convolved and rectified. The convolutional layer is composed of several convolved feature maps. Then pooling is applied and the result is several pooled feature maps. Then, the data are flattened to form the input of the ANN. The ANN combines the relevant features it detected on the previous steps and tries to classify the image.

5.7.2 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are advanced algorithms used mainly in time series analysis. In time series analysis, historical data are processed to form predictions for the future. The neurons in a RNN own some kind of short-term memory which make them capable of remembering the information that was in a neuron formerly. In that way, information can flow from the neurons to themselves in the future. That makes these networks suitable for tasks where historical information should be considered, such as natural language processing and speech recognition.

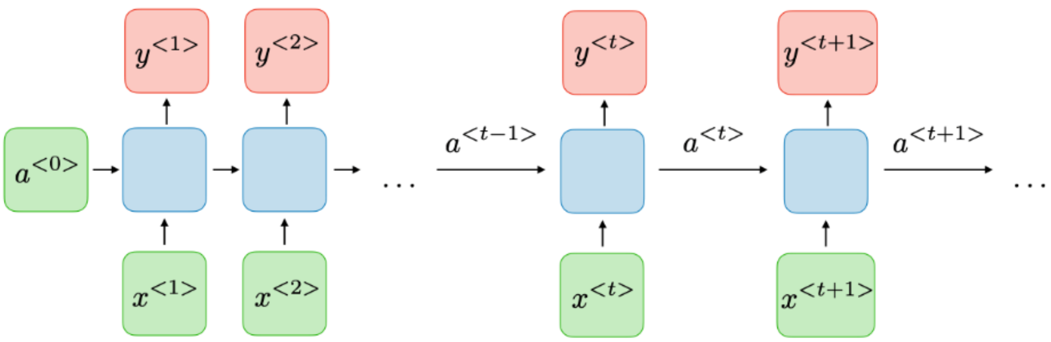


Figure 5.21 Architecture of a traditional Recurrent Neural Network [46]

On the Figure 5.21 a traditional RNN is represented. For each timestep t, the activation $\alpha^{<t>}$ and the output $y^{<t>}$ are expressed as follows:

$\alpha^{<t>} = g_1(W_{aa}\alpha^{<t-1>} + W_{ax}x^{<t>} + b_a)$ and $y^{<t>} = g_2\left(W_{ya}\alpha^{<t>} + b_y\right)$

where $W_{ax}, W_{aa}, W_{ya}, b_a, b_y$ are temporally shared coefficients and $g_1, g_2$are activation functions.

Vanishing Gradient Descent

As was explained, gradient descent is an algorithm whose aim is to minimize a cost function to find the optimized parameters of the network. The network's training happens as information flows through the network from input to output and the calculated error flows back in backpropagation process to alter the network's parameters.

The process in a RNN is quite similar as in an ANN but there are differences which can be found to the fact that the information from previous timesteps can be used to the next ones as it travels through time. That also means that the calculation of the cost function can occur at each timestep. All weights of the neurons that take part in the calculation of that cost function must alter, which results to the alteration not only of the weights of the neurons directly combined with the output, but also of the previous ones. But two problems occur when training a RNN with backpropagation. These are *vanishing gradient descent* and *exploding gradient* that happen when dealing with long-term dependencies [47] [48]. The issue is that some neurons of the network will not be trained properly. But since the output of a neuron is used as an input to further neurons, then the further layers will be trained from untrained ones. Finally, the entire network will not be trained properly [49]. If the weight used to connect the hidden layers to themselves is small that leads to vanishing gradient descent. If it is large, then exploding gradient issue occurs.

Possible solutions to exploding gradient may be the interruption of backpropagation after a specific point, the setting of a maximum threshold to the gradient or the artificial reduction of it. As for the vanishing gradient problem properly initialization of the weights, use of Echo State Networks and *Long Short-Term Memory Networks* may lead to a solution [50].

Long Short-Term Memory

Long Short-Term Memory network is a variation of a recurrent neural network and is used to overcome the issue of vanishing gradient descent [51].

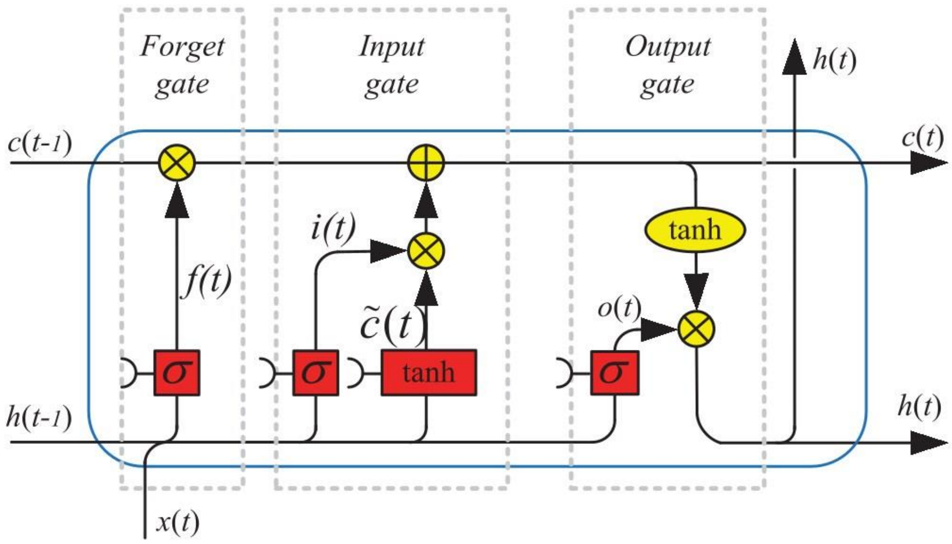


Figure 5.22 Architecture of LSTM model [52]

The Figure 5.22 illustrates the architecture of a LSTM model. The straight line on the top of the figure is called a memory cell and it flows through time, while on occasion it may be removed, or information may be added on. The term $c(t-1)$ is referred to the input from a memory cell in time t, $x(t)$ is the input in time t, $h(t)$ is the output in time that goes to the output layer and the hidden layer in the next timestep. So, every block at the network has three inputs which are $x_t, h(t-1)$ and $c(t-1)$ and two outputs $c(t)$ and $h(t)$. All these are not just single values but vectors. On the figure above there are shown some pointwise operations in yellow circles or ellipses. First, the x represents the forget, memory and output valves. These valves allow information to flow from that pipeline, to interrupt the flow, or allow it to flow but to some certain extent. For example, the forget valve in the top left corner of the figure will be closed, opened to some extent of fully opened. Fully opened means that the value of the forget gate $f(t)$ is 1 and keeps all the information as the memory flows from $c(t-1)$ through the pipeline to $c(t)$. Memory from $c(t-1)$ stops flowing in if the valve is closed and likely fresh memory will be added to the pipeline with another pointwise operation. The "+" pointwise operation adds new memory to the pipeline if the sigmoid layer operation σ opens the memory valve in the middle of the figure. The role of $tanh$ operation is to transform the values of the vector to the range from -1 to 1 to make them suitable for further processing.

In this paragraph the process of the LSTM block will be briefly explained. First, $x(t)$ and $h(t)-1$ from the previous block enter the block. These vectors are concatenated and passed through the sigmoid layer operation which opens or not the forget valve. The two vectors are then forwarded to the $tanh$ operation where it is decided with the memory valve what information will be passed to the pipeline. After that, information flows to the pipeline on the top either through the forget gate and/or the memory gate. If the forget valve is open and the memory valve is closed then the memory remains the same, while if the forget valve is closed and the memory valve open then, the memory will be fully updated. The concatenated vectors $h(t-1)$ and $x(t)$ are forwarded to the output valve where it is decided what information from the pipeline exits the block as output and to what extent depending on the value $o(t)$.

6 Framework for CSI-based indoor localization with 2D CNN

This section contains a detailed description of the entire process, from database creation to location estimations. Briefly, the database content, preprocessing, the fingerprint construction, the CNN model's architecture will be described and the results will be discussed.

6.1 Experiment Setup and Database Description

The data was collected by Ariadne Maps GmbH, a company that provides customized data-driven solutions for industries and crowd analytics services. The data collection was carried out during the offline phase as follows. In a small room at the company's faculties two devices were placed diagonally at a stable location on two of the corners of the room and were used as the receivers while one more device was used as the transmitter and was moved in 6 distinct locations in the room. The devices used, were the sensors ESP32 from ESPRESSIF. At each location the device was left for some time so that some packets were collected by the receivers and then was moved to the next location. In the next Figure 6.1, the locations at which data were collected are represented as green dots.

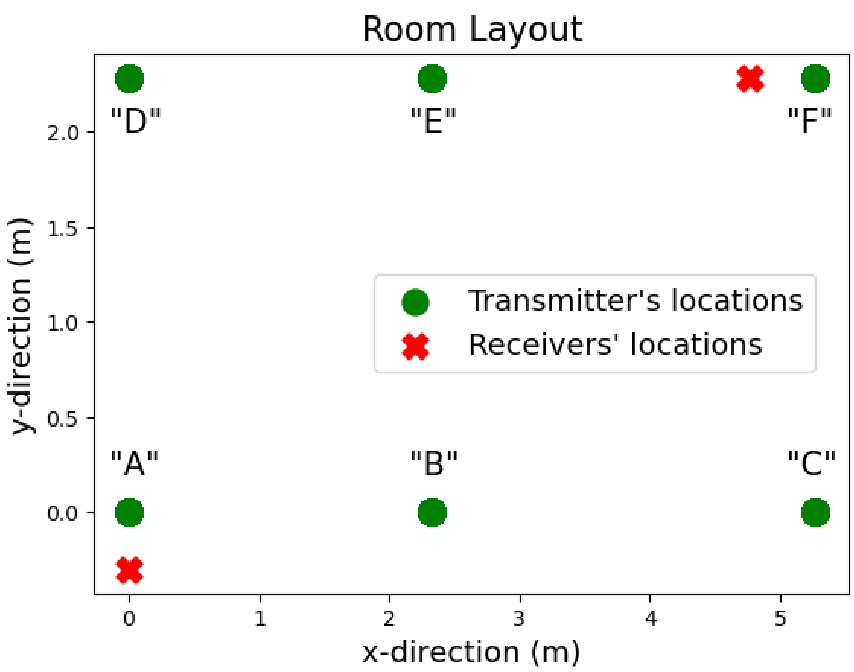


Figure 6.1 Transmitter's and receivers' locations in the experimental room

The database contains a large number of observations for each one of the six locations. Some of the features included in each observation are, among others, the timestamp, the CSI and RSSI data, the longitude and latitude of the transmitter and the receivers, the MAC address of the transmitter and finally the ID of the receivers.

This work is focused on the CSI data and for that reason uses the CSI data and the longitudes and latitudes of the receivers and the transmitter.

6.2 CSI fingerprint and database construction

A crucial part of the whole procedure is to create fingerprints for each location. These are also called CSI images. The fingerprint of one location should be distinguishable from a fingerprint that belongs to another location so better results may be produced. Each location may have more than one fingerprints but these may be similar to each other. The goal is to create a database that contains the CSI fingerprints and the corresponding (x, y) coordinates for each fingerprint. The Figure 6.2 illustrates graphically the basic concept of how the database is organized. Each fingerprint is associated with a location. The number N refers to the total number of packets collected from all locations and does not mean that there are N different locations. Some of the pairs (x_N, y_N) are identical.

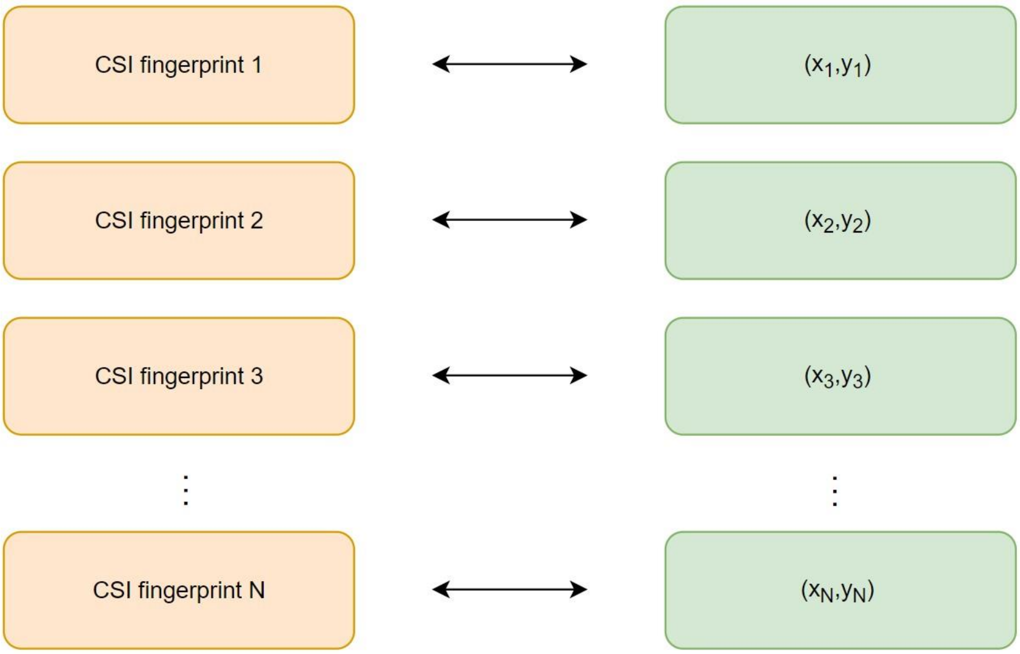


Figure 6.2 Illustration of database content

The next paragraph will describe the process of the CSI fingerprint construction and what it includes. The original database consists of multiple packets with CSI in their raw form.

Each CSI packet is a sequence of multiple numbers which by two are the real and the imaginary parts of a complex number $a + bj$. From each pair of the complex number's components contained in the sequence of each CSI packet, the amplitude and phase can be calculated. There are as many subcarriers as the number of pairs of a, b components in the sequence of a CSI packet. Finally, from each packet the values of amplitude and phase across all subcarriers are calculated. Amplitude and phase can be calculated by the following equations.

Since CSI is a complex number, it can be represented in different forms such as Cartesian and Polar ones. The Cartesian form consists of the real and imaginary components a and b, respectively, that have been mentioned previously. The polar one is made up of magnitude and phase. The first two equations show the two representations, polar and cartesian, respectively, while the 3rd and 4th equation show the conversion from cartesian to polar form.

$$CSI = |Mag|\angle\varphi$$

$$CSI = Re + iIm$$

$$Mag = \sqrt{Re^2 + Im^2}$$

$$\varphi = \arctan\,(Re, Im)$$

A critical fact is that two receivers are included in the experiment, which means that at each location packets from both receivers exist in the CSI fingerprint if that packets belong to the same or close timestamp. A CSI fingerprint is a 3D matrix that contains information for amplitude and phase across all subcarriers for both receivers. A graphical explanation of how the CSI fingerprint is constructed is shown in Figure 6.3. In the left part of the figure there are the matrices of the amplitude and phase constructed from a packet that receives the Receiver 1 and a packet from Receiver 2 from a specific location of the transmitter. The amplitude matrix has dimensions $N_R x\ N_C$ where N_R is the number of receivers and N_C is the number of subcarriers. The two matrices are then concatenated to produce a new 3D matrix which is actually the CSI fingerprint at a specific location. The 3D matrix has, finally, dimensions $N_R x\ N_C x 2$.

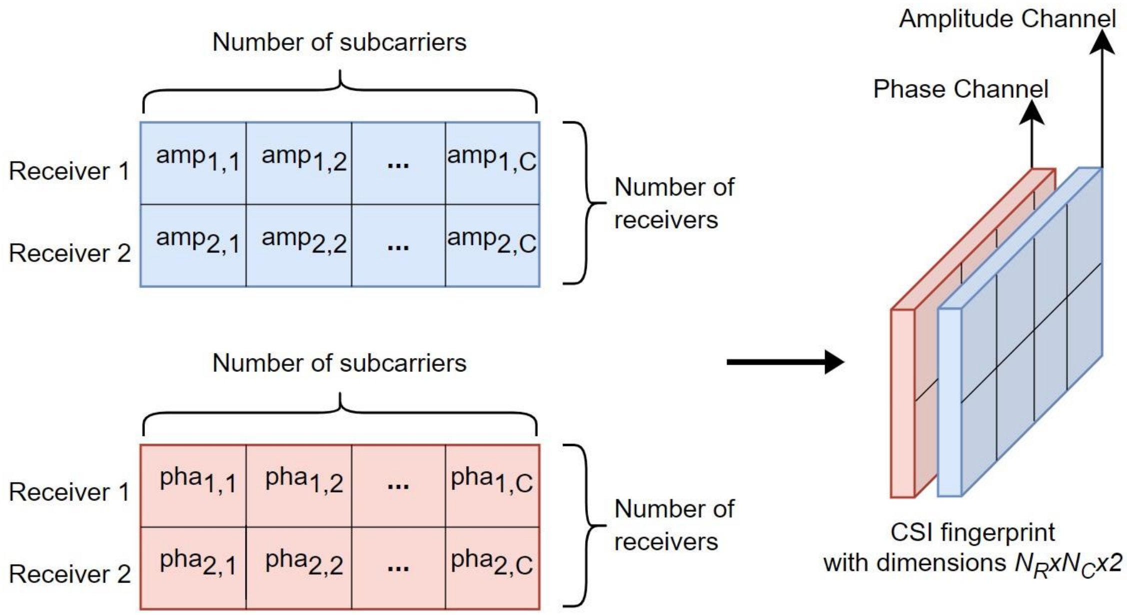


Figure 6.3 Conversion of amplitude and phase array to CSI fingerprint

Multiple fingerprints may refer to a single location and finally, after the construction of CSI fingerprints from all the available packets for all locations they are stored in the fingerprint database with the corresponding transmitter's coordinates. The resulting CSI fingerprints are used next as inputs in the CNN.

6.3 Data preprocessing

Before training the CNN with the constructed fingerprints, denoising techniques and normalization were considered. Hampel filtering was applied to the amplitude of each fingerprint in the database separately for each one of the two receivers. The technique of phase alignment described in the [14] was also applied to the phase information. The next figures illustrate, as an example, several packets of amplitude of two different locations, A and F, before and after preprocessing. Some of the peaks in the signals have been eliminated. It is also shown how distinct are the amplitude signals of the two locations.

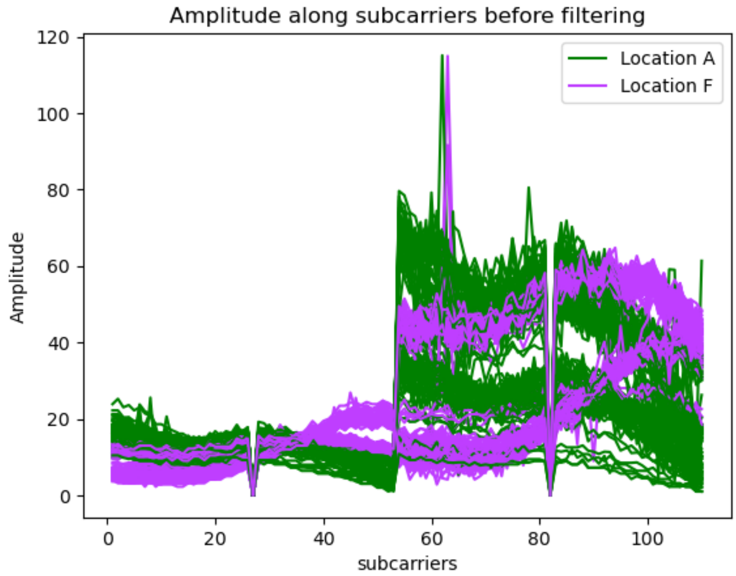


Figure 6.4 Amplitude signals before filtering for locations A and F

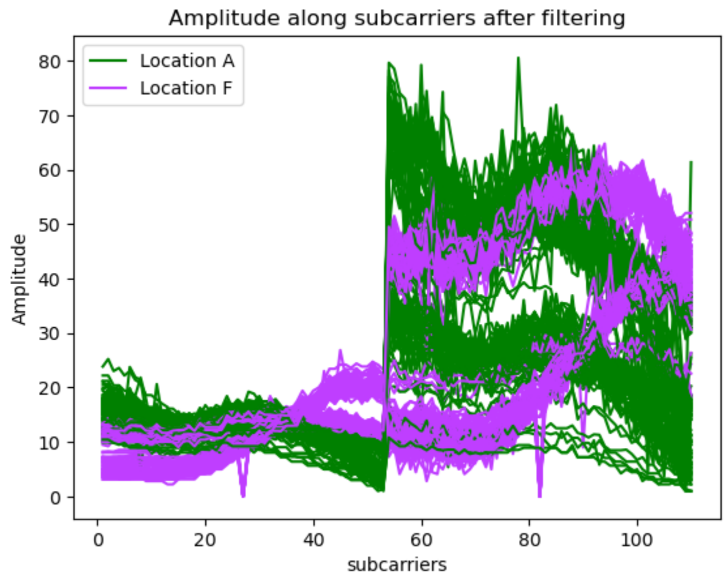


Figure 6.5 Amplitude signals after filtering for locations A and F

6.4 2D CNN-based Indoor localization

After the construction of the database and the preprocessing step of the amplitude and phase components, the CSI fingerprint database is used to train a Neural Network. The total number of packets was 1485 while 80% of that was used for training and the rest for testing. After the training of the network the coordinates estimation of the test CSI fingerprint are produced and compared to the ground truth. The Euclidean distance between the network's predictions and the actual values is calculated and the average distance is used to measure the model's performance.

6.4.1 Network architecture

The proposed 2D CNN architecture is illustrated in the Figure 6.6. The input is a CSI fingerprint with dimensions $2x110x2$ as explained in the section explaining the CSI fingerprint construction for the 110 subcarriers while the output is the estimation of the (x, y) coordinates. The proposed architecture uses two convolutional layers followed by max pooling layers. The size of the kernels in the convolutional layers is 2x2 and 32 filters are used while the size of the pooling matrix in the pooling layers is 1x3. The Rectified Linear Unit is used as the activation function and Adam is used as the optimizer. There are also three fully connected layers with 128 units each. Finally, the output layer has 2 units for the (x, y) coordinates.

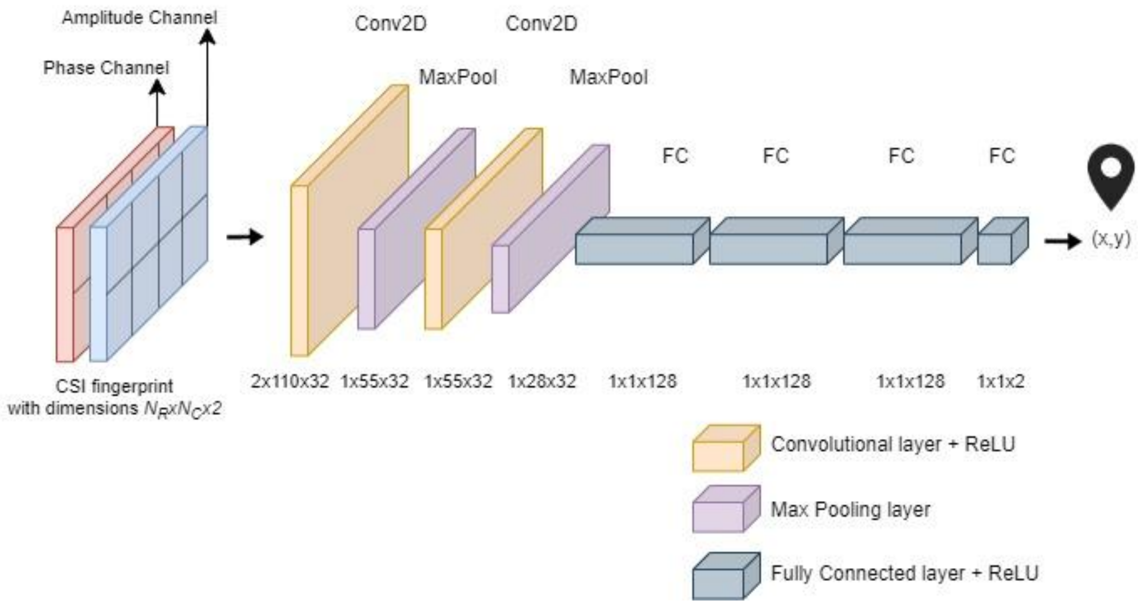


Figure 6.6 Convolutional Neural Network architecture

6.4.2 Experimental Results

This section presents the experimental results using the architecture of the mentioned CNN on the test set. The test set is a number of samples of CSI fingerprints with the corresponding coordinates. The coordinates of each sample may refer to one of the 6 points shown in the figure presenting the room layout.

The Figure 6.7 illustrates the actual target values of the test samples as green dots while the predictions of the coordinates in the test set are shown as red dots. In the test set there are only 6 unique pairs of (x, y) coordinates which means that a specific (x, y) pair may appear multiple times in the test set. The black arrows shown connect the estimations of the CSI fingerprint in the input and the ground truth of the corresponding fingerprint. Most of the predictions are concentrated around the ground truth. Each one of the 6 ground truth locations is perimetrically enclosed by multiple predictions. The closer to the ground truth the more concentrated the predictions are. That means that the model achieves a very good performance as it is able to predict the location of the fingerprints closely to the actual locations. These results were obtained using both amplitude and phase and the average distance error was about 0.53m. However, there are some predictions that are far away from the actual locations. That can be shown from the long arrows that connect some red points that are scattered far away from their ground truth. That fact can be explained by the presence of outliers in the corresponding CSI fingerprints that cause abnormal predictions.

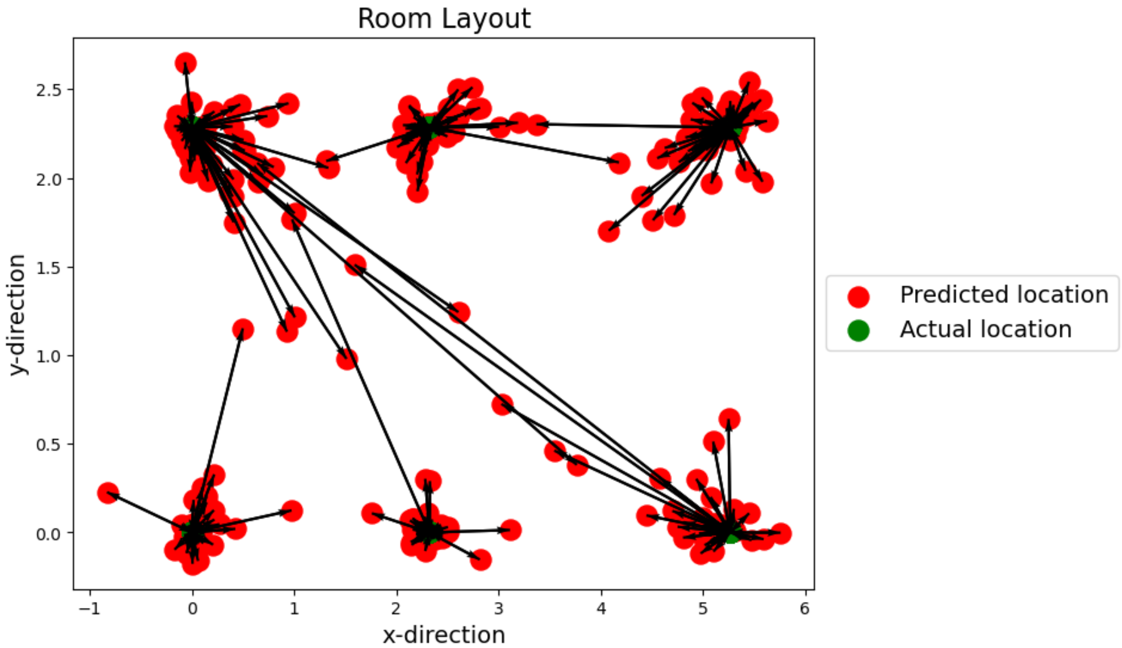


Figure 6.7 Estimated locations for the CSI fingerprints

Figure 6.8 presents the histogram of occurred errors in meters on Euclidian distance, x and y direction between the actual location and the predicted ones. The vast majority of the errors are concentrated between 0 and 1 meter, while in distances greater than 3 meters only a few occurrences appear for Euclidean and x direction as there is no error greater than 2.5 for y direction.

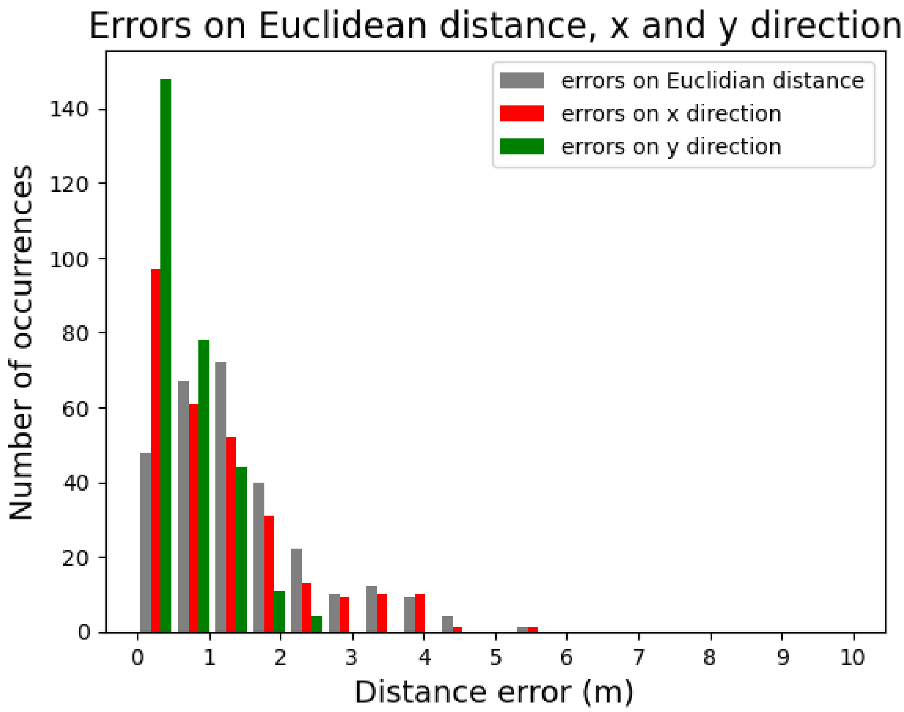


Figure 6.8 Distance error occurrences on Euclidean distance, x and y direction

The performance of the model can be evaluated in Figure 6.9 which presents the Cumulative Distribution Functions (CDFs) of distance errors on Euclidean distance, x and y directions between the actual and estimated locations. For the Euclidean distance more than 90% of the test fingerprints have an error under 1m. Almost the same occurs for the error on x direction. The CDF curve of the error on x direction is almost identical to the curve on Euclidean distance with a slightly better performance. Regarding the CDF on y direction at least 95% of the test fingerprint have an error under 1m, which is profoundly better than the performance of Euclidian distance and x direction. There is a simple explanation for that. The dimensions of the room and also the distance between the actual locations in y direction is about 2.5 meters. As a result, the error will be much less than that on Euclidean distance or x direction where actual locations appear along around 5.5 meters, which is almost the room's length. For that reason, the CDF which refers to the y direction stops at around two meters while for Euclidian and x direction continues to 5 meters distance error.

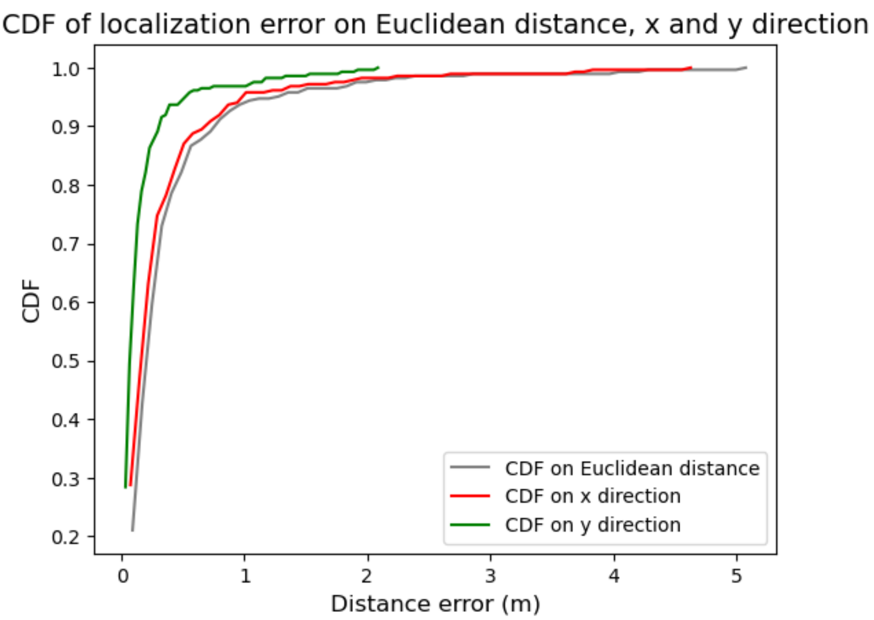


Figure 6.9 Cumulative Distribution Function of distance error

In Figure 6.10 the CDF is presented in three different cases. Specifically, the model's performance is compared in the case of using amplitude and phase, only amplitude and only phase. The results show that the worst performance is obtained when using only the phase of the fingerprints, while obviously better results are achieved when using both amplitude and phase or only amplitude. The use only of the amplitude achieves slightly better performance than using both amplitude and phase.

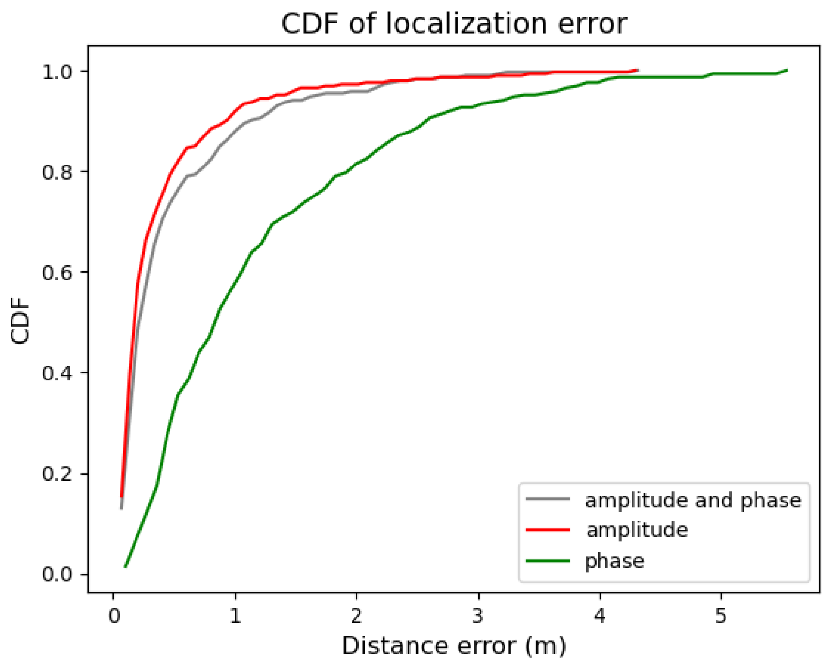


Figure 6.10 Cumulative Distribution Function of distance error using amplitude and phase, only amplitude and only phase

That is probably because the use of the phase in the CSI images does not produce as distinguishable fingerprints as in the case of using only the amplitude which means that the amplitude produces more stable fingerprints or better preprocessing and noise reduction is required to be applied on the phase channel.

7 Conclusions and Future Work

In that thesis, a framework for CSI based localization was designed. First, a thorough investigation into fundamental communication principles has been made. Additionally, a wide range of indoor localization technologies and techniques were presented. The working principle of several types of artificial neural networks was also discussed.

Furthermore, the experimental setup was described. Specifically, in a small room two devices were placed diagonally at a stable location on two of the corners of the room and were used as the receivers while one more device was used as the transmitter and was moved in 6 distinct locations in the room. At each location, the device was left for some time so that some packets were collected by the receivers and then was moved to the next location. The data was collected by Ariadne Maps GmbH which is a company that provides customized solutions for industries and crowd analytics services.

The next step was the construction of the fingerprints. Each fingerprint consisted of amplitude and phase information and each location had multiple fingerprints which may be similar to each other. The resulting CSI fingerprints are used next, after preprocessing, as inputs to a CNN. Convolutional layers and FC layers are combined to enhance the depth of the NN and increase localization accuracy while retaining complexity at a manageable level. The goal was to create a database that contains the CSI fingerprints and the corresponding (x, y) coordinates for each fingerprint.

After the training of the network, the coordinates estimation of the test CSI fingerprints are produced and compared to the ground truth. The Euclidean distance between the network's predictions and the actual values is calculated and the average distance is used to measure the model's performance.

The Cumulative Distribution Functions (CDFs) of distance errors on Euclidean distance, x and y directions between the actual and estimated locations was calculated. The CDF curve of the error on x direction is almost identical to the curve on Euclidean distance with more than 90% of the test fingerprints having an error under 1m. Regarding the CDF on y direction at least 95% of the test fingerprint have an error under 1m, which is profoundly better than the performance of Euclidian distance and x direction because the dimension of the room in y direction and also the distance between the actual locations is much less than that on Euclidean distance or x direction.

Finally, the model's performance is compared in three different cases, meaning the case of using amplitude and phase, only amplitude and only phase. The worst performance was obtained when using only the phase of the fingerprints, while obviously better results are achieved when using both amplitude and phase. Slightly better performance was achieved by using only amplitude. The use of the phase in the CSI images, probably, did not produce as distinguishable fingerprints as in the case of using only the amplitude. That may mean that the amplitude produces more stable fingerprints, or the phase channel requires more preprocessing and noise reduction.

The CSI-based localization topic is new, which opens up a lot of opportunities for potential future development. As future work, the use of an 1D CNN will be considered and the problem will be handled both as regression and classification task while different preprocessing approaches will be compared. Lastly, CSI information will be collected for more locations in a bigger space.

Bibliography

[1] M. W. Pasha *et al.*, “Enhanced fingerprinting based indoor positioning using machine learning,” *Computers, Materials and Continua*, vol. 69, no. 2, pp. 1631–1652, 2021, doi: 10.32604/cmc.2021.018205.

[2] N. C. Syazwani, N. Haliza Abdul Wahab, N. Sunar, S. H. S Ariffin, K. Yinn Wong, and Y. Aun, “Indoor Positioning System: A Review.” [Online]. Available: www.ijacsa.thesai.org

[3] Z. Li, K. Xu, X. Wang, H. Wang, Y. Zhao, and M. Shen, “Machine-learning-based positioning: A survey and future directions,” *IEEE Netw*, vol. 33, no. 3, pp. 96–101, May 2019, doi: 10.1109/MNET.2019.1800366.

[4] L. Wang and S. Pasricha, “A Framework for CSI-Based Indoor Localization with 1D Convolutional Neural Networks.”

[5] X. Song *et al.*, “A Novel Convolutional Neural Network Based Indoor Localization Framework with WiFi Fingerprinting,” *IEEE Access*, vol. 7, pp. 110698–110709, 2019, doi: 10.1109/ACCESS.2019.2933921.

[6] N. Singh, S. Choe, and R. Punmiya, “Machine Learning Based Indoor Localization Using Wi-Fi RSSI Fingerprints: An Overview,” *IEEE Access*, vol. 9. Institute of Electrical and Electronics Engineers Inc., pp. 127150–127174, 2021. doi: 10.1109/ACCESS.2021.3111083.

[7] J. M. Rocamora, I. W. H. Ho, W. M. Mak, and A. P. T. Lau, “Survey of CSI fingerprinting-based indoor positioning and mobility tracking systems,” *IET Signal Processing*, vol. 14, no. 7. Institution of Engineering and Technology, pp. 407–419, Sep. 01, 2020. doi: 10.1049/iet-spr.2020.0028.

[8] Y. Rezgui, L. Pei, X. Chen, F. Wen, and C. Han, “An Efficient Normalized Rank Based SVM for Room Level Indoor WiFi Localization with Diverse Devices,” *Mobile Information Systems*, vol. 2017, 2017, doi: 10.1155/2017/6268797.

[9] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, “Recurrent Neural Networks For Accurate RSSI Indoor Localization,” Mar. 2019, doi: 10.1109/JIOT.2019.2940368.

[10] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey, *DeepFi: Deep Learning for Indoor Fingerprinting Using Channel State Information*. 2015.

[11] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach," in *IEEE Transactions on Vehicular Technology*, Institute of Electrical and Electronics Engineers Inc., Jan. 2017, pp. 763–776. doi: 10.1109/TVT.2016.2545523.

[12] X. Wang, X. Wang, and S. Mao, "Deep Convolutional Neural Networks for Indoor Localization with CSI Images," *IEEE Trans Netw Sci Eng*, vol. 7, no. 1, pp. 316–327, Jan. 2020, doi: 10.1109/TNSE.2018.2871165.

[13] C. H. Hsieh, J. Y. Chen, and B. H. Nien, "Deep Learning-Based Indoor Localization Using Received Signal Strength and Channel State Information," *IEEE Access*, vol. 7, pp. 33256–33267, 2019, doi: 10.1109/ACCESS.2019.2903487.

[14] A. Foliadis, M. H. Castañeda Garcia, R. A. Stirling-Gallacher, and R. S. Thomä, "CSI-based localization with CNNs exploiting phase information," in *IEEE Wireless Communications and Networking Conference, WCNC*, Institute of Electrical and Electronics Engineers Inc., 2021. doi: 10.1109/WCNC49053.2021.9417561.

[15] K. Wu, J. Xiao, Y. Yi, M. Gao, and L. M. Ni, "FILA: Fine-grained indoor localization," in *Proceedings - IEEE INFOCOM*, 2012, pp. 2210–2218. doi: 10.1109/INFCOM.2012.6195606.

[16] J. Xiao, K. Wu, Y. Yi, and L. M. Ni, "FIFS: Fine-grained Indoor Fingerprinting System."

[17] Y. Chapre, A. Ignjatovic, A. Seneviratne, and S. Jha, "CSI-MIMO: Indoor Wi-Fi fingerprinting system," in *Proceedings - Conference on Local Computer Networks, LCN*, IEEE Computer Society, Oct. 2014, pp. 202–209. doi: 10.1109/LCN.2014.6925773.

[18] H. Obeidat, W. Shuaieb, O. Obeidat, and R. Abd-Alhameed, "A Review of Indoor Localization Techniques and Wireless Technologies," *Wireless Personal Communications*, vol. 119, no. 1. Springer, pp. 289–327, Jul. 01, 2021. doi: 10.1007/s11277-021-08209-5.

[19] F. Zafari, A. Gkelias, and K. Leung, "A Survey of Indoor Localization Systems and Technologies," Sep. 2017, [Online]. Available: http://arxiv.org/abs/1709.01015

[20] X. Feng, K. A. Nguyen, and Z. Luo, "A survey of deep learning approaches for WiFi-based indoor positioning," *Journal of Information and Telecommunication*, vol. 6, no. 2, pp. 163–216, 2022, doi: 10.1080/24751839.2021.1975425.

[21] J. Wang and J. G. Park, "An enhanced indoor positioning algorithm based on fingerprint using fine-grained csi and rssi measurements of ieee 802.11n wlan," *Sensors*, vol. 21, no. 8, Apr. 2021, doi: 10.3390/s21082769.

[22] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Comput Surv*, vol. 46, no. 2, 2013, doi: 10.1145/2543581.2543592.

[23] C. Yang and H. R. Shao, "WiFi-based indoor positioning," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, Mar. 2015, doi: 10.1109/MCOM.2015.7060497.

[24] A. Nessa, B. Adhikari, F. Hussain, and X. N. Fernando, "A Survey of Machine Learning for Indoor Positioning," *IEEE Access*, vol. 8, pp. 214945–214965, 2020, doi: 10.1109/ACCESS.2020.3039271.

[25] R. F. Brena, J. P. García-Vázquez, C. E. Galván-Tejada, D. Muñoz-Rodriguez, C. Vargas-Rosales, and J. Fangmeyer, "Evolution of Indoor Positioning Technologies: A Survey," *Journal of Sensors*, vol. 2017. Hindawi Limited, 2017. doi: 10.1155/2017/2630413.

[26] S. J. Hayward, K. van Lopik, C. Hinde, and A. A. West, "A Survey of Indoor Location Technologies, Techniques and Applications in Industry," *Internet of Things (Netherlands)*, vol. 20. Elsevier B.V., Nov. 01, 2022. doi: 10.1016/j.iot.2022.100608.

[27] M. A. Al-Ammar *et al.*, "Comparative survey of indoor positioning technologies, techniques, and algorithms," in *Proceedings - 2014 International Conference on Cyberworlds, CW 2014*, Institute of Electrical and Electronics Engineers Inc., Dec. 2014, pp. 245–252. doi: 10.1109/CW.2014.41.

[28] J. Kunhoth, A. G. Karkar, S. Al-Maadeed, and A. Al-Ali, "Indoor positioning and wayfinding systems: a survey," *Human-centric Computing and Information Sciences*, vol. 10, no. 1. Springer, Dec. 01, 2020. doi: 10.1186/s13673-020-00222-0.

[29] K. (Kevin N.) Gurney, *An introduction to neural networks*. UCL Press, 1997.

[30] S. Hitziger and S. H. Modeling, "Modeling the variability of electrical activity in the brain." [Online]. Available: https://theses.hal.science/tel-01175851

[31] B. C. Csáji, “Approximation with Artificial Neural Networks Approximation with Artificial Neural Networks Huub ten Eikelder.” [Online]. Available: https://www.researchgate.net/publication/357888472

[32] S. Sharma, S. Sharma, and A. Athaiya, “□ACTIVATION FUNCTIONS IN NEURAL NETWORKS,” 2020. [Online]. Available: http://www.ijeast.com

[33] “https://medium.com/.”

[34] L. Bottou, “Stochastic Gradient Learning in Neural Networks.”

[35] “A list of cost functions used in neural networks, alongside applications.”

[36] Y. Zhang, P. Fitch, M. P. Vilas, and P. J. Thorburn, “Applying multi-layer artificial neural network and mutual information to the prediction of trends in dissolved Oxygen,” *Front Environ Sci*, vol. 7, no. MAR, 2019, doi: 10.3389/fenvs.2019.00046.

[37] J. Grus, “Data Science from Scratch.”

[38] S. Ruder, “An overview of gradient descent optimization algorithms,” Sep. 2016, [Online]. Available: http://arxiv.org/abs/1609.04747

[39] “https://developer.nvidia.com/.”

[40] “https://www.mltut.com/.”

[41] C. Gershenson CGershenson, “Artificial Neural Networks for Beginners.”

[42] P. H. Sydenham and R. Thorn, *Handbook of measuring system design*. Wiley, 2005.

[43] “Application of Digital Image Processing to Marker-free Analysis of Human Gait,” 2001, doi: 10.13140/2.1.1046.7847.

[44] D. Podareanu, V. B. Codreanu, S. Bv, S. Aigner, V. Weinberg, and G. Caspar Van Leeuwen, “Best Practice Guide-Deep Learning Czech-Bavarian Competence Team for Supercomputing Applications (CzeBaCCA) View project DFG Forschergruppe FOR 465: Gitter-Hadronen-Phänomenologie View project Best Practice Guide-Deep Learning,” 2019, doi: 10.13140/RG.2.2.31564.05769.

[45] M. Suriya, V. Chandran, and M. G. Sumithra, “Enhanced deep convolutional neural network for malarial parasite classification,” *International Journal of Computers and Applications*, 2019, doi: 10.1080/1206212X.2019.1672277.

[46] V. Maeda-Gutiérrez *et al.*, “Comparison of convolutional neural network architectures for classification of tomato plant diseases,” *Applied Sciences (Switzerland)*, vol. 10, no. 4, Feb. 2020, doi: 10.3390/app10041245.

[47] “https://stanford.edu/.”

[48] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training Recurrent Neural Networks,” Nov. 2012, [Online]. Available: http://arxiv.org/abs/1211.5063

[49] Yoshua Bengio, Patrice Simard, and Paolo Frasconi, “Learning Long-Term Dependencies with Gradient Descent is Difficult,” 1994.

[50] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” 2013.

[51] A. Karpathy, J. Johnson, and L. Fei-Fei, “Visualizing and Understanding Recurrent Networks,” Jun. 2015, [Online]. Available: http://arxiv.org/abs/1506.02078

[52] Sepp Hochreiter and Jurgen Schmidhuber, “Long Short-Term Memory,” 1997.

[53] Y. Yu, X. Si, C. Hu, and J. Zhang, “A review of recurrent neural networks: Lstm cells and network architectures,” *Neural Computation*, vol. 31, no. 7. MIT Press Journals, pp. 1235–1270, Jul. 01, 2019. doi: 10.1162/neco_a_01199.