GRADUATING INTERNSHIP

Signal processing using Neural Networks for the physical layer of 5G

Student:
Mona DHIFLAOUI

Company supervisor:
M. Guillaume LARUE

School supervisor:
M.Laurent ROS

An internship report submitted in fulfillment of the requirements for the degree of SICOM Engineering

In

Grenoble INP - Phelma

July 8, 2020

"Life can be much broader once you discover a simple fact, and that is — everything around you that you call life, was made up by people that were no smarter than you. And you can change it, you can influence it, you can build your own things that other people can use. Once you learn that, you'll never be the same again."

Steve Jobs

Abstract

Grenoble INP - Phelma
Orange Labs

SICOM Engineering

Signal processing using Neural Networks for the physical layer of 5G

by Mona DHIFLAOUI

The fifth generation (5G) wireless communications are expected to manage the fleet of massive devices with real-time communication and a large amount traffic data in a complex scenarios. That is why the future 5G networks will require robust intelligent algorithms, such as Artificial intelligence (AI) algorithms, to manage resources and networks in different situations.
The physical layer is the key element that ensure the ultra-reliability and low-latency traffic transmission. In this context, we choose the aim of this project to be the use of Low Complexity Neural Networks to perform base band equalization operations. We proposed the idea of presenting the analytical function for both cases one-tap (path) and multi-tap equalization, by a simple neural network that gives the same results as the normal equalizer.
The task will be done on several stages; first, it will concern one-tap channel equalization by understanding the conventional equalizer and then proposing a NN that will replace it. Second it will be for multi-tap channel equalization by choosing two different techniques of equalization which are The linear method MMSE and non-linear method; The Decision Feedback Equalizer methods replace them by a suitable architecture of NN.
The proposed NN models are compared with classic algorithm. Results shows similar BER performances which was expected. The purpose of using low complexity NN in this project is not to have better performances, but is to ensure higher efficiency and reduced learning and inference time.

La cinquième génération de réseaux mobiles 5G promet d'améliorer les services et les applications en offrant au réseau la qualité de fonctionnement, la fiabilité et des débits se mesurant en Gigabit/s et en supportant la connexion simultanée d'un plus grand nombre d'appareils (IOT). En effet, les opportunités technologiques qu'elle laisse entrevoir sont vastes et demandent des algorithmes robustes et intelligents. L'utilisation d'algorithmes basés sur l'IA ou le ML sont identifiés comme des candidats potentiels pour répondre à ces défis.
La couche physique de réseau s'occupe de la transmission physique des données. Elle constitue le premier élément de la chaîne de traitement du signal d'un récepteur et sa qualité est cruciale pour palier au contraintes de débit, de fiabilité, de la latence ou encore d'efficience énergétique d'un système de communication. L'objectif de l'étude suivante est de proposer des algorithmes intelligents comme les réseaux de neurones pour les intégrer au fonctionnement de cette couche.
Pour la 5G appliquée dans l'IOT, les notions comme la complexité et la consommation d'energie ont une grande importance. Pour cela, nous avons proposé l'utilisation des réseaux de neurones à complexité controlée comme une réponse potentielles aux contraintes de l'IoT pour égaliser le canal avec différentes configurations (one-tap et multi-tap). La première partie du travail est consacrée au canal one-tap et propose une architecture de réseau de neurones simple permettant de remplace la méthode traditionnel d'égalisation. La deuxième partie décrit différentes méthodes d'égalisation multi-trajets ainsi que leurs solutions équivalentes sous la forme de réseaux de neurones à faible complexité.
Ce type de réseau de neurones avec faible complexité montrent une performance similaire à celle des méthodes classique, ce qui nous donne l'espoir d'avoir des nouveaux technique rapide et avec faible consommation d'énergie pour être appliqué dans l'IOT.

Acknowledgements

"To me, you've done your best."

I would first like to thank my supervisors Mr.Guillaume Larue, Mr.Louis-Adrien Dufrene and Mr.Quentin Lampin in Orange Labs. The door to their office was always open whenever I ran into a trouble spot or had a question about my research or writing. Mr.Guillaume Larue consistently allowed this paper to be my own work, but steered me in the right direction whenever he thought I needed it and he gave me the honor to participate in the preparation of his research paper "Low-Complexity Neural Networks for Physical Layer".
I would like to acknowledge Professor Ros Laurent for his insightful comments and encouragement. It was a great privilege and honor to work and study under his guidance. I am extremely grateful for what he has offered me.

Some special words of gratitude go to my friend Paul Aimé who has always been a major source of support when things would get a bit discouraging.
Finally, I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study. this accomplishment would not have been possible without them.

Contents

List of Figures

List of Abbreviations

IOT	**I**nternet **O**f **T**hing
M2M	**M**achine to **M**achine
PHY	**Phy**sical
AI	**A**rtificial **I**ntelligence
ML	**M**achine **L**earning
NN	**N**eural **N**etwork
CNN	**C**onvolution **N**eural **N**etwork
DFE	**D**ecision **F**eedback **E**qualizer
DNN	**D**ense **N**eural **N**etwork
TPU	**T**ensor **P**rocessing **U**nit
UDN	**U**ltra **D**ensification **N**etwork
MIMO	**M**ulti **I**nput **M**ulti **O**utput
mmWave	**m**illimeter **W**ave
NLP	**N**atural **L**anguage **P**rocessing
CV	**C**omputer **V**ision
ASR	**A**utomatic **S**peech **R**ecognition
ISI	**I**nter **S**ymbol **I**nterference
DFE	**D**ecision **F**eedback **E**qualizer

Chapter 1

Introduction and context

1.1 Introduction

The new generation of wireless communication needs to explore the abilities of Artificial intelligence (AI) to be able to support the massive devices with real-time communication in an ultra-reliable way and with low-latency. In fact, the groundbreaking success in computer vision, natural language processing and other fields, encourages us to think that the AI can be able to intelligently manage the large amount of traffic data and to deal with complex scenarios of communication systems.

In this context, this project introduces the use of comprehensive concepts of Artificial intelligence (AI), in general, and artificial neural networks (ANNs), in particular, and their potential applications in the physical layer of the 5G and beyond. The physical layer is a key element to ensure a reliable transfer of huge data, but the rate of data transmissions is limited due to the effects of linear and nonlinear distortion such as inter symbol interference (ISI) in the presence of additive white Gaussian noise. Different equalization techniques (Zero forcing, Wiener filter, Decision Feedback Equalizer, etc.) are introduced to reduce or even to remove those effects.

For this purpose, we present a comprehensive overview on a number of key types of neural networks that include dense and convolutional. Those neural networks will be a tools to replace the conventional techniques of equalization. For each type of neural network, we present the basic architecture and training process, as well as the related challenges and opportunities. Those ideas will allow us to think about communications system design as sequences of neural networks.

1.2 Outline of the report

Chapter one clarifies the project environment by presenting the company and the team of the project and gives an overview about the problem statement, the aim and the methodology of the work. In chapter two, origins, fundamentals and methods of ANN are discussed and a background to the equalization techniques that will be studied in this project is introduced. Chapter three presents the descriptions and the instruction of neural networks to equalize the channel in different configuration. In chapter four, the main conclusions of this project are discussed.

1.3 Work environment

1.3.1 Company

Orange is one of the leader in the market of telecommunication services to businesses, mobile and broadband internet services with 236 million customers worldwide and 42.2 billion Euros in 2019 revenues.
Orange Labs is the Research and Development division of Orange. It serves to identify the technological breakthroughs, to use them in decision making process for business. Their mission is to make all of the expertise of Orange Labs available to develop competitive solutions for the business customers all around the world.

1.3.2 Presentation of the team

My work in this internship was within the CITY team in Orange Labs, Meylan. It is a group of 31 persons, 8 of them are PHD students, 4 are experts of future networks. The mission of this team is mainly focused on IoT and M2M systems (Smart Cities). The project team during this internship is comprised of 2 Research engineers, Louis-Adrien Dufrene (PhD), Quentin Lampin (PhD), and a PhD Student, Guillaume Larue (MSc). Since 1 year, they started the investigation of the opportunities offered by AI and ML for signal processing at the physical (PHY) layer.

1.4 Motivation and Objectives

Thanks to its optimization capabilities, Artificial intelligence (AI) has been applied recently for many fields, such as computer vision and natural language processing. The potential extension of AI to the physical layer has also become increasingly recognized due to the latest constraints for future communication systems, such as dynamic situations with ambiguous channel models, high speed and precise processing requirements; which challenge traditional communication theories.
In fact in the case of AI applied to the signal processing of the physical layer, we need to distinguish between the application of the ML and the NN. The common idea that we all have about ML is that it needs a lot of data to get a good performance, but using NN for this case doesn't need a lot of data to be trained because it can be seen as a mathematical bridge between the analytical function and the new one. Moreover the high-cost hardware of telecommunication systems is one of the reasons that push us to think about using AI, because the dedicated hardware for Neural Network(NN) models(Neural Processing Unit NPU)is available with low cost and it can consume less energy compared with traditional Signal Processing Equipment that are extremely specific.This NPU can be parameterized with any kind of NN.

This project provides a detailed overview of the evolving AI-based physical layer processing studies, including leveraging ML to redesign a section of the conventional communication system which is the equalizer and replace it with a fundamentally new architecture based on a Neural Network.
In fact this project is a continuation of some of the work of the team that has led to the redaction of a conference paper, *"Low-complexity Neural Networks for physical layer"* (Currently under reviewing) in which I had the chance to participate. These paper describes two models to realize single-path equalization and demodulation of M-QAM signals. These models are assessed using both simulation and experimentation and achieve near optimal performances.

To sum-up, the continuity of the work in this internship will focus on leveraging Low-Complexity NN to redesign the conventional channel equalizer for two different types of channel; one-tap and multi-tap channel in the aim of using intelligent algorithms to resolve a complex scenarios of communication systems with low energy consumption.

1.5 Project management

To ensure a solid path to project success, we need a clearly established planning (Figure 1.1) with fixed goals to achieve the final aim. The first two months were reserved to understand the project and to participate in the work for the research paper which was the one-tap equalization model and its test using USRP and Channel emulator.In April we focused on the multi-tap equalization, we started by studying the different techniques in the conventional communication theories to choose a technique that will be used in the rest of project which is the Wiener filter that will be done in the month of May. Finally, we have finished the project by studying an other method of multi-tap equalization which is the Decision Feedback Equalizer(DFE).

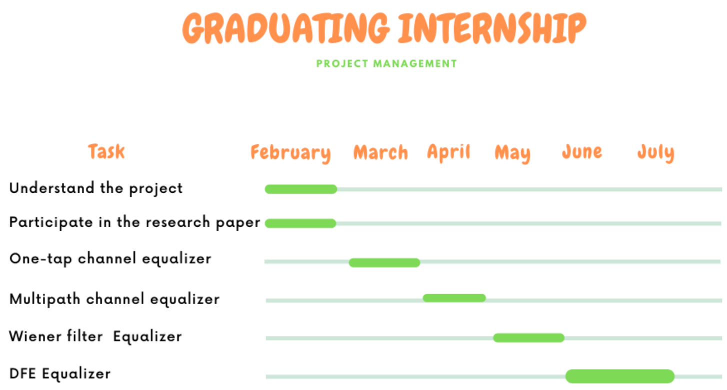


FIGURE 1.1: Gantt chart of the project

The project aims at developing elements of a NN-based physical layer processing and using NN to reinvent a section of the traditional communication system (which is in our case the channel equalization) and replace the communication system block with a fundamentally new concept based on models of Neural Networks.

Chapter 2

State of art

2.1 Introduction

This chapter will provide, in its first section, a detailed overview about ANN modeling by introducing its different types. The second section will introduce the motivations and the goal to use AI for the physical layer. Third section will present the problem of the channel equalization in different configurations, as well as theoretical solutions.

2.2 Neural Network

The artificial neuron (AN) is the basic element of the artificial neural network (ANN). It is inspired from the biological neuron and its function. The biological neuron is composed of dendrites, soma, axon, and synapses. Dendrites receive inputs from other neurons, which are in some way combined in soma. Axon perform a generally non-linear operation, and then outputs are produced in the synapses which are the final stage of the neuron cell.

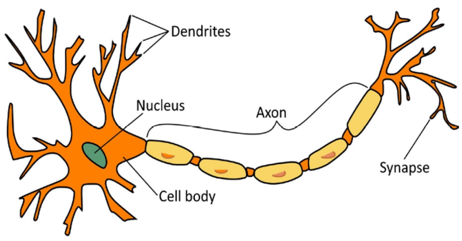


FIGURE 2.1: A biological neuron

Biological neuron is a specific cell. It controls abilities like identification, thinking by introducing a connection with other neurons and applying previous experiences to every action. This ability is based on the interconnection between a huge number of the biological neurons that allows it to learn from experience that is why a single cell (neuron) is useless.

The artificial neuron uses mathematical function to create a simplified model of a human neuron . Each input enters the neuron will be multiplied by its specific connection weights that represent the importance of the corresponding input in the output. Then, they are summed and processed by an activation function in order to generate output.

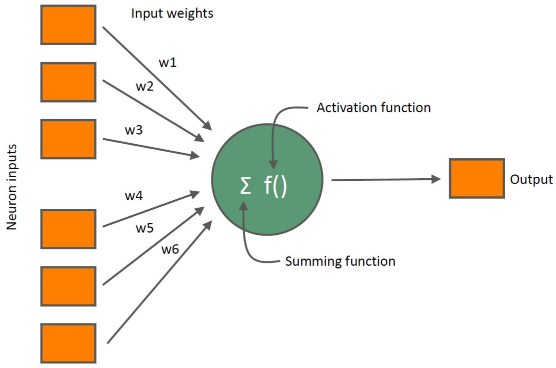


FIGURE 2.2: Artificial neuron

Artificial Neural Networks(ANNs) is no more than an interconnection of artificial neurons. ANNs learn, from the previous experiences, the relations between selected inputs and outputs. The ANNs attempt to form a rough parallel to the way that neurons function in the human brain.[19].
ANNs can be very fast comparing to the conventional algorithms by exploiting the parallelism because they can perform their tasks simultaneously.

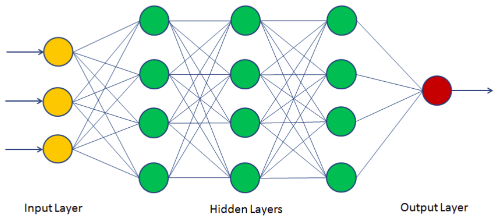


FIGURE 2.3: Artificial Neural Network (ANN)

Neural networks can be classified as feed-forward, recurrent, convolutional, modular, Radial Basis Function Neural Network ,etc. In this project, we choose to work with dense and convolutional neural networks.

2.2.1 Dense Network

A dense neural network consists of a series of fully connected (dense) layers. Each neuron receives input from all of the previous layer's neurons, and is thus densely connected.

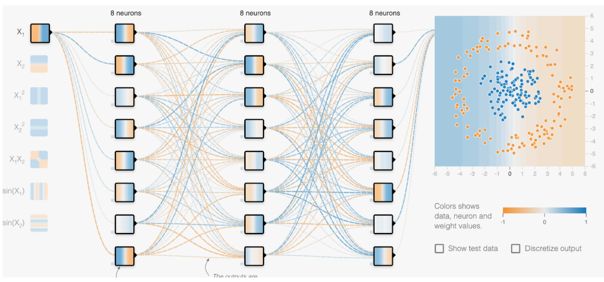


FIGURE 2.4: Dense Neural Network Representation on TensorFlow Playground

The layer has a matrix of weight W, a bias vector b and the outputs of the previous layer. The output of each layer is formed as follows:

$$output = f\left(\sum_{j=0}^{n} \omega_j x_j + b\right) \quad (2.1)$$

Where $x_j$represents the input data, the kernel represents the weight data, $\sum_{j=0}^{n} \omega_j x_j$ the sum of weighted inputs, b is the bias value used in machine learning to optimize the model and f is the activation function[8]. To sum up,dense layer is a linear operation in which each input is related by a weight to each node in the following layer. Generally followed by a non-linear activation function.

2.2.2 CNN

Convolutional neural networks (CNNs) are a kind of feed forward neural network where each single node can be used to apply filters across overlapping regions[12]. For dense neural networks, we need a preprocessing for the data to extract the data and use them as input for the network. However, in the case of data with high dimensions, more parameters will be required which makes the work with dense networks more complicate. CNNs have been proposed to tackle this issue. The CNNs depend on weight sharing, which reduces parameter counts.
A local filter (kernel) is applied to the input image by the convolution layers. By applying the filter to an image subset, certain local features will be removed. By eventually merging them, we can create the same format as the original image but with less dimensional data.
CNN is often applied for images processing. It applies a filter that slides over the input image to create a feature map. The convolution of another filter gives a different feature map over the same image that is why the more number of filters we have, the more image features get extracted. CNN learns the values of these filters on its own during the training process.

For example in the figure below, we have an image of size (4,4,5) where the third dimension represents the number of channels (for example it can represents the RGB channel in the case of image (nxnx3)). The convolutional layer will apply on it a filter with a kernel size (1,5) which means the same filter is applied on the 5 channels. The output of this process is an image with a reduced dimension which contains the extracted features of the input image[12].

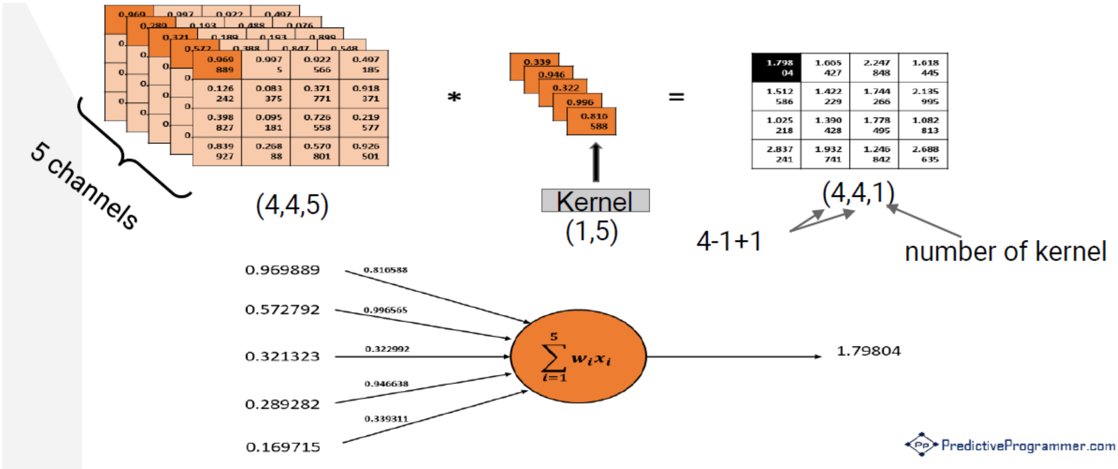


FIGURE 2.5: Convolutional neural networks [12]

To sum up, CNNs are a special class of neural networks. They have proven very effective in areas such as image recognition and classification. In fact if we take the vision application as example, we find that they generate low-dimensional reproduction that yields outstanding numerical results in classification, detection, scene understanding, etc.

Neural Network consists of an assortment of algorithms used in Machine Learning to model data using neuron graphs. Whereas, Machine Learning uses sophisticated algorithms that read, learn from the data, and use them to discover interesting patterns.

2.2.3 Machine Learning

Machine learning (ML) is an artificial intelligence (AI) sub-category which provides systems with the ability to learn and improve automatically from experience without being explicitly programmed. Machine learning focuses on developing computer programs which can access data and use it to learn on their own. Its task is aimed at identifying (learning) a function f mapping the input domain X (of data) to the output domain Y (of potential predictions). The functions f are chosen from various groups of functions, depending on the type of learning algorithm used. Machine Learning (ML) and Artificial Intelligence (AI) are two very hot buzzwords right now. However, there is a confusion regarding what truly is artificial intelligence, and what truly is machine learning [19]. So, it is worth to explain the difference.

- ***Artificial Intelligence(AI)***: Any methodology allowing computers to perform tasks normally requiring human intelligence; using logic, if-then rules, machine learning and decision trees.
- ***Machine learning(ML)***:A subset of AI that integrates statistical techniques that allow machines to improve on experienced tasks.This category encompasses deep learning
- ***Deep learning(DL)***: it is a category of machine learning algorithms which is based on neural networks with more than three layers (i.e. more than one hidden layer), So called Deep Neural Netwrks (DNN).

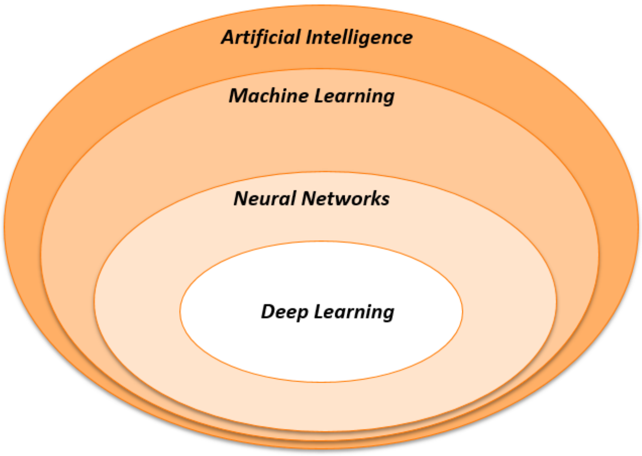


FIGURE 2.6: AI vs ML vs DL

Depending on the properties of the used data-sets, Machine learning algorithms can generally be divided into three groups: Supervised learning, unsupervised learning and reinforcement learning.
The dataset of **the supervised learning** is a *labeled dataset* $(x, y) \in \mathbf{X} * \mathbf{Y}$, where x represents a data point and y the corresponding true prediction for x. This input-output pair training set is used to find a deterministic function that maps every input to an output, predicting future input-output observations while eliminating as much as possible errors.
To train the system, **unsupervised learning** systems use *unlabeled datasets*. The aim of unsupervised learning is to derive structure from unlabeled data by investigating the similarity between pairs of objects, and is usually associated with estimating density or clustering of data.
Reinforcement learning systems do not have a fixed data set, but instead a feedback loop between the system and its experience[4]. The goal of reinforcement learning is to map situations into actions with the objective of optimizing rewards.
There are other learning systems which are a mixture of two types, Such as **semi-supervised learning**, using both labeled and unlabeled knowledge[2].

There are a wide range of tasks that can be solved by machine learning. A regression analysis and classification are two common machine learning tasks.
Regression is one of the most important and commonly used methods for machine learning and statistics.It is used to make data predictions by learning the relationship between the data features and some observed, continuous-evaluated responses. This task is solved by outputting a function $f : \mathbb{R}^n \mapsto \mathbb{R}$ that fits the data. For example, the regression analysis can be used to predict future stock prices in the trading environment.
In **classification** the computer is asked to decide the group G that a certain input belongs to. The task can be solved by outputting a function $f : \mathbb{R}^n \mapsto \{1, ..., n\}$. A common classification problem is object recognition for intelligent systems.

To evaluate which machine learning methodology is suitable for a particular application, one may analyze the aspects required for an optimally supervised machine learning pipeline to be built. Kotsiantiset al.[9] describes a pipeline that can be used to construct a good classifier for new data instances that generalizes well. This pipeline is illustrated in the figure below.

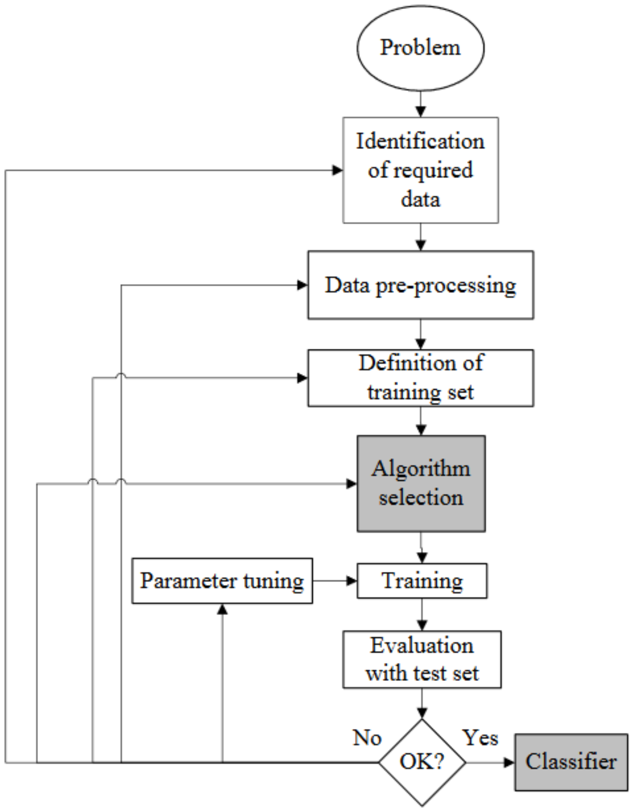


FIGURE 2.7: The process of supervised ML [9]

The first two stages of this pipeline are the most significant, and they essentially determine the classifier's performance. The identification of necessary data includes the determination and selection of the most important features. The data dimensionality can be reduced by excluding irrelevant or redundant features. Too much information that is irrelevant or redundant may prevent a learning algorithm from finding patterns in data, or even result in false results. The pre-processing phase is used to deal with this redundancy of information, and is also used to fix noise or missing values. The input of the training dataset is the final product of the pre-processing phase[9].

2.3 AI for physical Layer

The impressive achievement of Machine learning (ML) in many fields such as; computer vision (CV), automatic speech recognition (ASR), and natural language processing (NLP),was the reason to gain the attention of the research community and the industry which are actively trying to expand these technologies to other areas, including wireless communication[18].
The integration of ML and NN theories into a wide range of communication systems has achieved many achievements since the 90s, especially in the upper layers such as cognitive radio, resource management[20], interconnection adaptation[16],[10]and positioning. And maybe, it is the time now to apply it for PHY layer.

The explosion in the number of connected IoT devices, to make the dream of smart cities comes true, has driven the development of wireless communications

into the fifth generation to face challenges like the necessity to ensure low-latency transmission, huge traffic management and massive connectivity.
Massive multi-input multi-output (MIMO), millimeter wave (mmWave), and ultra-densification network (UDN) are some examples of innovative solution to face the challenges of the advanced wireless applications but they are not enough to resolve the complex scenarios[18]. That is why the idea of using AI in the wireless communications was introduced in the hope of resolving those limitations. Some reasons of using AI in physical Layer are listed as follows[18]:

- *Modelling complex scenarios* : Channels are designed using a mathematical models to describe actual environments. However, some cases struggle with many imperfections and nonlinearities which makes the modeling difficult and that what we call by complex scenarios. Therefore, systems or algorithms that can complete communication tasks without defined channel models are essential.
- *The need for reliable and fast signal processing*: The new signal processing for wireless communication needs effective and fast treatments but, unfortunately, the traditional dedicated hardware (DSP, FGPA, ASIC) is no more efficient and it will introduce additional nonlinear imperfections in those complex scenarios. In the other hand, the arrival of efficient programming frameworks and efficient processing platform (GPU, TPU,NPU), made the AI, a potential effective solution and fast solution for complex signal processing. Moreover those dedicated hardwares are available with low-cost and with diversity in the market.
- *Limitation of the traditional algorithms*: the traditional algorithms are too complex or energy-hungry to be applied to restricted devices in their conventional form (polar codes as an example). Moreover, the existing works on the AI demonstrate that it can provide similar performance with lower complexity.

For those reasons, researchers applied ML to the physical layer for many applications such as recognition of modulation[1], [3], modeling and detection of channels[14], [7], encoding and decoding[5], estimating channels and equalization[17], [6]. In this project as mentioned before, we will show the results and the achievements of applying the ANNs for the equalization task of the PHY layer.

2.4 Equalization

In wireless communication, many distortions affect the signal transmitted over a channel (such as the fading channel,noise,Doppler shift,etc) which provoke fluctuations in amplitude and phase. The delay spread of the channel introduces also inter-symbol interferences (ISI) to the received signal[11].
In fact, Multipath is a signal propagation phenomenon which describe the case where the signals enter the receiving antenna by two or more routes. ISI can introduce a high error levels if not compensated.

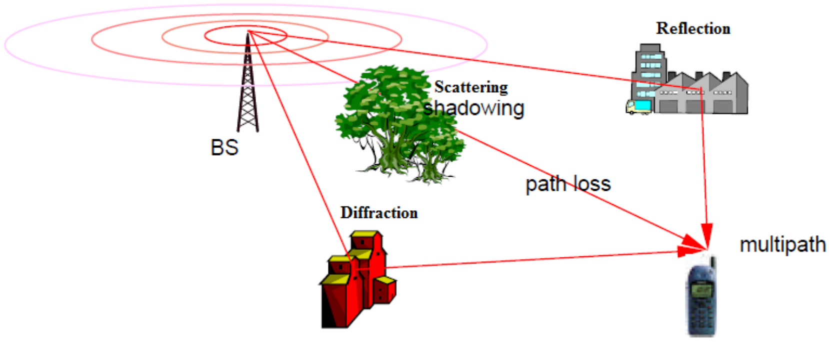


FIGURE 2.8: Multipath channel in wireless communications[11]

Obviously, reducing the effects of ISI is crucial for a reliable digital transmission system and that is where the equalizers come onto the scene.

One of the efficient method of wireless channel modeling is to represent the channel as a band-limited digital filter which will be defined with transfer function. Thus, in the purpose of alleviate and reduce the channel effects especially in the case of the multi-path fading and ISI channel, a digital filter with transfer function can be designed based on the inverse of the transfer function of the associated wireless channel. This digital filter is called the equalizer[15]. So, channel equalization is a process by which the channel's effects on the transmitted signal is compensated and the resulting ISI reduced or even removed.
An equalizer or method of equalization can be incorporated in the compositions of the receiver block as shown in the figure below. Different equalization techniques can be used to mitigate the ISI.

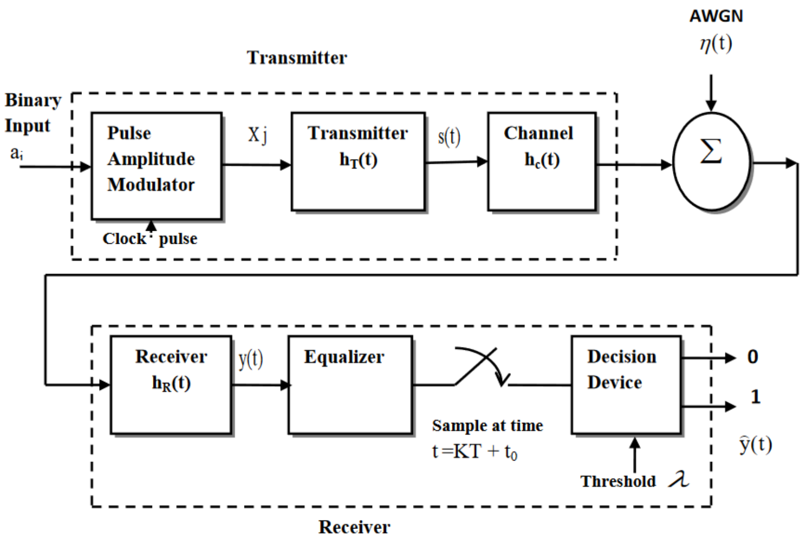


FIGURE 2.9: Block diagram of a digital communication system

Techniques of equalisation fall into two broad categories: linear and nonlinear. The linear methods are usually the easiest to conceptually apply and understand. Linear equalization methods, however, usually suffer from noise enhancement and hence are not used in most wireless applications. We can cite as examples of Linear equalization techniques; Zero-forcing method, MMSE(Weiner filter),etc. Decision-feedback equalization (DFE) is one of the most common method among nonlinear equalization techniques as it is relatively simple to implement and does not suffer from noise enhancement[13].

2.4.1 One-tap

The single path channel is the case of a channel with one single dominant path (strong propagation). The channel adds a delay and attenuation to the transmitted signal.

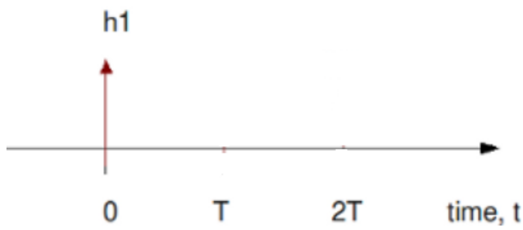


FIGURE 2.10: Single-path channel model

The transmit signal through a one-tap channel is given as:

$$y(t) = x(t) * h_1(t) \tag{2.2}$$

This kind of channel is considered as the ideal scenario of channel configuration because of the absence of the inter symbol interference (ISI). It is certainly a simple model but nevertheless widely used in modern systems based on OFDM.

2.4.2 Multi-tap

The transmission of the signal X, through a multi-path channel, results a multiple replicas of X that will be received (signal Y) at the receiver. To study the effect of this multi-path channel, we need to understand the relationship between the transmitted, received signal and the channel model. The multi-path channel can be represented as a band-limited digital filter with a transfer function H.

As an example, the three-path channel can be described as it is shown below.

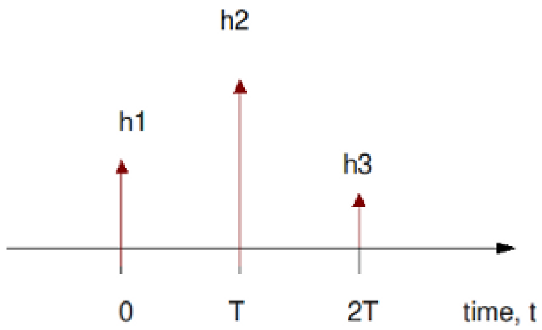


FIGURE 2.11: Channel model (3 tap multi-path)

The transmitted symbols can be modeled as:

$$y(t) = X(t) * h(t) = \sum_{n=0}^{L} x_n h(t - nT) \tag{2.3}$$

Where x_n is the symbol to transmit, $h(t)$ is the transmit filter, n is the symbol index, L is the length of the transmit filter and T is the symbol index.
An other modeling representation of this relationship can be introduced using toeplitz matrix: The received signal **Y** is given by:
$Y = H.x_{(1)} = X.h_{(2)}$

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ \vdots \\ y_m \\ \vdots \\ \vdots \\ \vdots \\ y_k \end{pmatrix} = \begin{pmatrix} h0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ h1 & h0 & 0 & \cdots & \cdots & \cdots & \cdots & \vdots \\ h2 & h1 & h0 & \ddots & \cdots & \cdots & \cdots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \cdots & \cdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \cdots & \vdots \\ \vdots & \cdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \cdots & \cdots & \ddots & h2 & h1 & h0 & 0 \\ \vdots & \cdots & \cdots & \cdots & \ddots & h2 & h1 & h0 \\ \vdots & \cdots & \cdots & \cdots & \cdots & 0 & h2 & h1 \\ 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 & h2 \end{pmatrix} \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ \vdots \\ x_m \\ \vdots \\ \vdots \\ x_{k-3} \end{pmatrix} \tag{2.4}$$

$$\begin{pmatrix} y_0 \\ y_1 \\ \vdots \\ \vdots \\ \vdots \\ y_m \\ \vdots \\ \vdots \\ \vdots \\ y_k \end{pmatrix} = \begin{pmatrix} x_0 & 0 & 0 \\ x_1 & x_0 & 0 \\ x_2 & x_1 & x_0 \\ x_3 & x_2 & x_1 \\ \ddots & \ddots & \ddots \\ \ddots & \ddots & \ddots \\ \ddots & \ddots & \ddots \\ x_{k-3} & x_{k-4} & x_{k-5} \\ 0 & x_{k-3} & x_{k-4} \\ 0 & 0 & x_{k-3} \end{pmatrix} \begin{pmatrix} h_0 \\ h_1 \\ h_2 \end{pmatrix} \tag{2.5}$$

The design of the transmitters and receivers depends on the assumption of the channel transfer function. However, in most of the digital communications applications, the channel transfer function is not known. Thus, an estimation of channel is requested to remove the channel effect at the receivers. For this task one of the techniques for channel estimation is the least-square (LS) channel estimation method. It minimizes the following cost function:

$$J(H_{estim}) = \|Y - Xh_{estim}\|^2 \tag{2.6}$$

By setting the derivative of the function with respect to H_{estim} to zero, we found that the LS channel estimation is:

$$H_{estim} = \left(X^H X\right)^{-1} X^H Y \tag{2.7}$$

In this equation we will use the matrix representation of X of the formula in the equation (2.10) of the convolution. The both techniques of equalization (Linear and nonlinear) can be used in the case of the multi-path channel.

Linear method: ***Zero Forcing Equalization***

The studies of this kind of equalizer will serve to check the performance of the wiener filter because they are similar in the high SNR regime. The ZF enforces a completely flat (constant) transfer function of the combination of channel and equalizer by choosing the equalizer transfer function as:

$$H_{ZF}(z) = 1/H(Z) \tag{2.8}$$

H_{estim} in this equation is the toeplitz matrix of the H_{estim} calculated by the LS in the section Channel Estimation.
The zero-forcing (ZF) technique nullifies the interference by the following weight matrix:

$$W_{ZF} = (H_{estim}^H H_{estim})^{-1} H_{estim}^H \tag{2.9}$$

The Zero Forcing Equalizer suffers from noise enhancement. It has a strong amplification at frequencies where the transfer function of the channel reaches small values, and thus also amplifies the noise.
The equalizer's purpose is to minimize not the ISI(Inter-Symbol Interference), but the probability of the bit error. Noise enhancement makes the ZF equalizer inappropriate for that.

Linear method: ***MMSE Equalization(Wiener filter)***

An other way to equalize the transmitted sympols X is the minimization of the Mean Square Error (MSE) between the transmit signal and the output of the equalizer$\hat{X}$. The idea is to find $\bar{C}$ that minimize the distance between the estimated and the transmit signal.
We choose that C such that:

$$minE\left\{\left\|\hat{X} - X\right\|^2\right\} = minE\left\{\left\|C^H Y - X\right\|^2\right\} \tag{2.10}$$

Note,

$$\begin{aligned}\mathbf{A} &= \left(C^H Y - X\right)\left(C^H Y - X\right)^H \\ &= \left(C^H Y - X\right)\left(Y^H C - X^H\right) \\ &= C^H Y Y^H C - C^H Y X^H - X Y^H C + X X^H\end{aligned} \tag{2.11}$$

We know that :

$$\begin{cases} auto-correlation \quad matrix \quad of \quad \mathbf{Y} \quad E\left\{YY^H\right\} & = R_{YY} \\ cross-correlation \quad matrix \quad of \quad \mathbf{XY}: \quad E\left\{XY^H\right\} & = R_{XY} \\ cross-correlation \quad matrix \quad of \quad \mathbf{YX}: \quad E\left\{YX^H\right\} & = R_{XY}^H \quad = R_{YX} \end{cases}$$

$$\begin{aligned} minE\left\{\mathbf{A}\right\} &= minE\left\{\left\|C^H Y - X\right\|^2\right\} \\ &= min\left(C^H R_{YY} C - R_{XY} C - C^H R_{YX} + R_{XX}\right) \\ &= min\left(C^H R_{YY} C - 2C^H R_{YX} + R_{XX}\right) \\ &= F(C)\end{aligned} \tag{2.12}$$

So,

$$\partial minE\left\{\mathbf{A}\right\} \setminus \partial C = \partial F(C) \setminus \partial C = 2R_{YY}C - 2R_{YX} \tag{2.13}$$

To find C we need that:

$$\partial F(C) \setminus \partial C = 0 \Leftrightarrow 2R_{YY}C - 2R_{YX} = 0 \Leftrightarrow R_{YY}C = R_{YX}$$

Thus:

$$\boxed{C = R_{YY}^{-1} R_{YX}} \tag{2.14}$$

$$\boxed{\hat{X} = C^H Y = R_{XY} R_{YY}^{-1} Y} \tag{2.15}$$

Now to find C we need to calculate R_{YY} and R_{XY}

$$R_{YY} = E\left\{YY^H\right\} = E\left\{(HX+N)(HX+N)^H\right\}$$

$$R_{YY} = E\left\{HXX^H H^H + NX^H H^H + HXN^H + NN^H\right\}$$

We assume that :

1. X is a sequence of *i.i.d* symbols with zero-mean and power p_d Where, $E\left\{XX^H\right\} = P_d\mathbf{I}$
2. X and N are independent ($E\left\{XN^H\right\} = 0 \Leftrightarrow E\left\{NX^H\right\} = 0$).

So,

$$\begin{aligned} R_{YY} &= E\left\{HXX^H H^H\right\} + E\left\{NX^H H^H\right\} + E\left\{HXN^H\right\} + E\left\{NN^H\right\} \\ &= E\left\{HXX^H H^H\right\} + E\left\{NN^H\right\} \\ &= HE\left\{XX^H\right\} H^H + \sigma_N^2 \mathbf{I}\end{aligned} \tag{2.16}$$

$$\boxed{R_{YY} = P_d HH^H + \sigma_N^2 \mathbf{I}} \tag{2.17}$$

R_{XY} is given by:

$$\begin{aligned} R_{YX} =& E\left\{Yx^H\right\} \\ =& E\left\{(HX+N)\,X^H\right\} \\ =& E\left\{HXX^H + NX^H\right\} \\ =& HE\left\{XX^H\right\} + E\left\{NX^H\right\} \end{aligned} \tag{2.18}$$

$$\boxed{R_{YX} = HP_d} \tag{2.19}$$

Thus,

$$C = R_{YY}^{-1} R_{YX} = \left(P_d HH^H + \sigma_N^2 \mathbf{I}\right)^{-1} P_d H \tag{2.20}$$

In the case of BPSK we have a sequence of symbols with mean = 0 and power $P_d = 1$ so,

$$C = R_{YY}^{-1} R_{YX} = \left(HH^H + \sigma_N^2 \mathbf{I}\right)^{-1} H \tag{2.21}$$

To sum up, the **MMSE weight matrix is given as**:

$$\boxed{C = \left(H_{estim} H_{estim}^H + \sigma_N^2 \mathbf{I}\right)^{-1} H_{estim}} \tag{2.22}$$

Note that the MMSE receiver requires the statistical information of noise σ_N^2
So the estimated sequence is given by:

$$\boxed{\hat{X} = C^H Y = R_{XY} R_{YY}^{-1} Y = H_{estim}^H \left(H_{estim} H_{estim}^H + \sigma_N^2 \mathbf{I}\right)^{-1} Y} \tag{2.23}$$

Note that:

$$H_{estim} = \left(X^H X\right)^{-1} X^H Y$$

Non-linear method:*Decision Feedback Equalizer(DFE)*

The decision-feedback equalizer (DFE) is nonlinear equalizer. It is composed of two filters, a feedforward filter and a feedback filter, arranged as shown in the figure below.

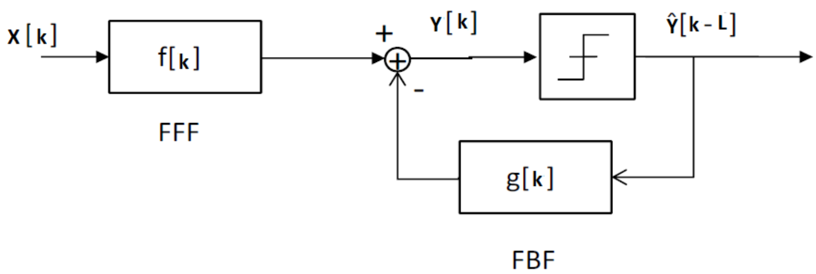


FIGURE 2.12: Structure of decision-feedback equalizer

Received signal sequence is the input to the feedforward filter f. Then, the decision sequence on the previously symbols will be the input of the feedback filter g which is used to delete the part of the ISI from the present estimated symbol. The DFE is nonlinear, since the detector feeds hard decisions to the feedback filter.
The equalizer output can be expressed in matrix-vector form as:

$$y[k] = F^T X_k - G^T \hat{y}_{k-L-1} \tag{2.24}$$

Where,$Y[n]$ depends on the previous and current symbols.

$$\begin{aligned} F &= [f_0, ..., f_{L-1}]^T \\ G &= [g_0, ..., g_{M-1}]^T \\ X_k &= [x[k], ..., x[k-L+1]]^T \\ \hat{y}_{k-L-1} &= [\hat{y}[k-L-1], ..., \hat{y}[k-L-M+1]]^T \end{aligned} \tag{2.25}$$

Note that L is the length of the feedforward filter and M-1 is the length of the feedback filter.
The basic elements of the DFE are the two filters and they need to be well chosen to ensure a good performance of the DFE. For that, we use MMSE criterion to find those filters($f_{opt}[n]$ and $g_{opt}[n]$). The idea is to minimize the cost function $J(w)$ which is the mean square error between the desired signal and the output of the equalizer. However, the DFE has same potential flaws such as error propagation which makes it a poor candidate for blinde qualization. Or,

$$\begin{aligned} J(w) =& E\left(|e[n]|^2\right) \\ e[n] =& y_k - X_k \end{aligned} \tag{2.26}$$

2.5 Conclusion

The aim of the project is to equalize the channel in different configuration using neural networks. This part of the report is an overview on neural networks and their different categories. Plus, it introduces the purpose of the equalization phase and its different techniques.

Chapter 3

Methodology and implementation

3.1 Introduction

In this chapter we propose an other vision of channel equalization using neural networks. The idea is to use the ANN as a tool that will take us to the same results as the conventional solution with low energy consumption and faster if we use the parallelism. For that, we will show the performance of ANNs in different channel configurations.

3.2 One-tap equalization

In this project we will study a sequence of M-QAM Modulated samples in the case of the following single path channel.

3.2.1 Conventional Solution

A random binary sequence is modulated using a Gray mapped M-QAM modulator which means they are in the complex space and to simplify their presentation, we choose to use the matrix notation instead of complex valued parameter. This sequences will be affected by the anomalies of the single-tap channel which can be:

- *Adding noise*: an Additive White Gaussian Noise (AWGN) will be added to the transmitted signal through the channel. This noise has a real and imaginary parts that are sampled from a zero-mean Gaussian distribution.

$$n = \begin{pmatrix} n_i & n_q \end{pmatrix}^T \; where \; n_i, n_q \sim N(0, \sqrt{\frac{N_0}{2}}) \tag{3.1}$$

- *Phase shift*: the phase error can be seen as a rotation in the complexe plane (*IQ* space). The symbols are rotated by an angle θ with the following rotation matrix.

$$R = \begin{pmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{pmatrix} \tag{3.2}$$

- *Unbalanced amplitude*: A different scaling factor applies to channels I and Q as defined in the diagonal stretching matrix below:

$$A = diag \begin{pmatrix} \alpha_i & \alpha_q \end{pmatrix} \tag{3.3}$$

- *Offset*: The *IQ* offset can be modelized by the vector:

$$o = \begin{pmatrix} \omega_i & \omega_q \end{pmatrix}^T \tag{3.4}$$

Note that x is a transmitted sample before being affected by the channel effects and y is the sample after the channel. The relationship between x and y is represented as follow:

$$y^T = x^T RA + (o+n)^T$$
$$\begin{pmatrix} y_i \\ y_q \end{pmatrix}^T = \begin{pmatrix} x_i \\ x_q \end{pmatrix}^T \begin{pmatrix} \alpha_i cos(\theta) & \alpha_q sin(\theta) \\ -\alpha_i sin(\theta) & \alpha_q cos(\theta) \end{pmatrix} + \begin{pmatrix} \omega_i + n_i \\ \omega_q + n_q \end{pmatrix}^T \tag{3.5}$$

The equalizer in the case of the one-tap channel will apply the inverse rotation operation to the received samples. So, the equalized sample will be represented as below:

$$u = \tilde{x} + \tilde{n}$$

Where u is the equalized sample, $\tilde{x}$ is the estimated sample and $\tilde{n}$ is the residual noise after the equalization. From the equation (2.6) we can represent the equalized sample as follows:

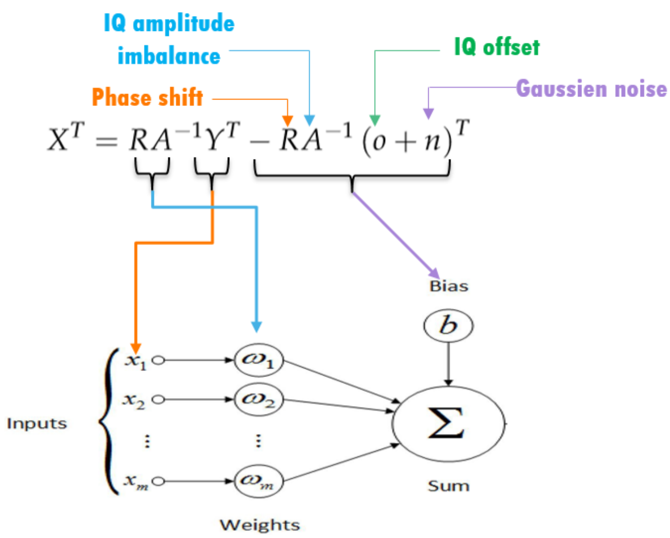
$$\begin{aligned} u^T &= (y-o)^T (RA)^{-1} \\ &= y(RA)^{-1} - o(RA)^{-1} \\ &= x^T - n^T (RA)^{-1} \end{aligned} \tag{3.6}$$

With

$$(RA)^{-1} = \frac{1}{\alpha_i \alpha q} \begin{pmatrix} \alpha_q cos(\theta) & -\alpha_q sin(\theta) \\ \alpha_i sin(\theta) & \alpha_i cos(\theta) \end{pmatrix}$$

3.2.2 NN Solution

The aim of this part is to find a way to represent the conventional solution of one-tap channel equalization as a neural network.
The shape of this analytical solution gives as the spark that we can use the neural network as a bridge to achieve the same results as the analytical techniques. Without forgetting that the dedicated hardware for ANNs are widely available in the market with low-cost and low-energy consumption which is very efficient especially in IOT applications.



It is clear from the above figure that a simple dense neural network is suitable to equalize the one-path channel. The samples are modulated using a Gray mapped M-QAM modulator, so they are in the complex space. Since, it is difficult to work with complex values in neural network, the input data will be split into two parts; real and imaginary parts. Thus the input data is a matrix with the shape (K,2) where K represents the number of the samples and 2 is the number of the two channels (real (I channel) and imaginary (Q channel) parts of these samples). We assume that the channel is static and it is a block-fading model, so, the samples are *i.i.d* and the NN can apply the equalization task independently to all samples affected by the same channel. Moreover, the NN is updated regularly (for each block/frame) with same online training process. For that two neurons and one layer is enough to equalize all the samples independently. The weight matrix is given by the matrix $(RA)^{-1}$.

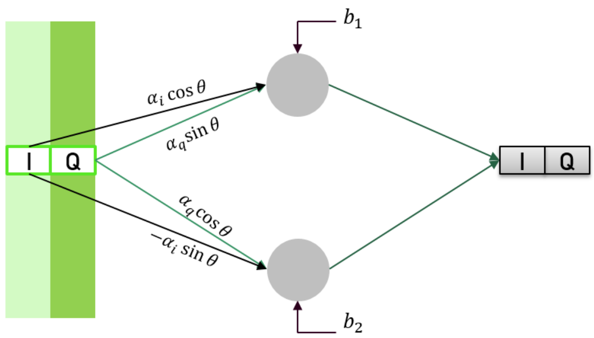


FIGURE 3.1: The proposed NN architecture for single-path equalization

The following results confirms that this simple model of NN was able to equalize the samples and to remove the channel effects. To get those results we used 8192 samples and 100 epochs with batch-size=10 but for real 200 samples for the same number of epochs was enough to get the same performance.

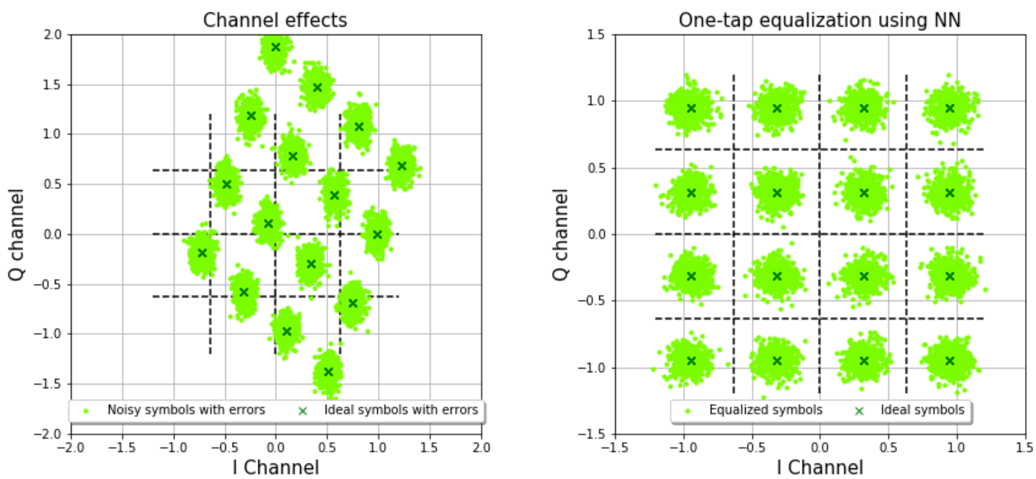


FIGURE 3.2: Left figure: Received samples with impairments - Right figure: Samples equalized by the NN.

3.2.3 Improvement

It turns out that we can reduce the number of parameters by using a CNN model in the place of a dense neural network in the case of using a high dimension data as input for the NN. The number of trainable parameters defines the complexity and the run-time of the model and that is very important for communications applications (i.e IOT). The CNN is very suitable because independent of the size of the input (1,10 or 10000 samples), it will always have 6 shared weights whereas the Dense NN will have a number of parameters proportional to the size of the input.

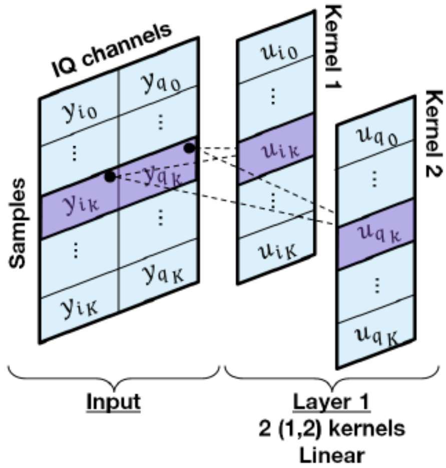
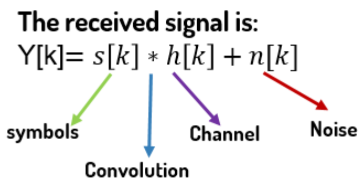


FIGURE 3.3: The proposed CNN architecture for single-path equalization has six shared parameters (two weights and one bias per kernel).

3.3 Equalisation Multi-tap

In the case of the multi-path channel, the received signal is defined as the convolution between the transmit signal and the transfer function of the channel.



The main idea in the equalization of the multi-path channel is to find the is the pseudo-inverse of the transfer function h.

3.3.1 Linear method:Wiener filter(MMSE)

The Wiener filter(MMSE) is one of the techniques that can find this pseudo-inverse of the transfer function h using the Minimum Mean Square Error (MMSE) criterion.

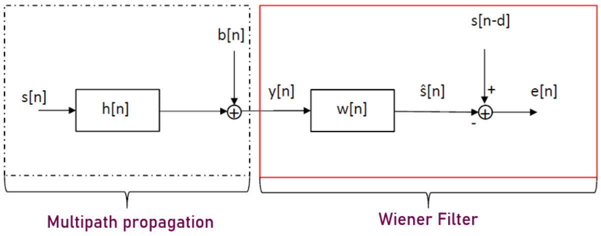


FIGURE 3.4: The diagram block of the Wiener filter method.

Conventional Solution

The calculation in the section 2.4.2 demonstrate that, using the Wiener filter method, the pseudo-inverse of the transfer function is given by the equation below:

$$C = R_{YY}^{-1} R_{YS} = \left(HH^H + \sigma_N^2 \mathbf{I}\right)^{-1} H \tag{3.7}$$

C can be calculated directly using the auto-correlation matrix of the received samples R_{YY} and the cross-correlation matrix R_{YS} or it can be calculated using the estimation of the channel.
To equalize the samples, we need just to convolve the received samples with the the pseudo-inverse C.

$$\hat{S} = C^H \otimes Y$$

NN Solution

The idea of equalizing the samples using the Wiener filter is based on finding the filter that will reduce or remove the multi-path channel effects. This idea pushes us to think about the CNN. In fact, this kind of NN allow us to find this pseudo-inverse filter without estimating the channel's transfer function. However, it needs just the estimation of the number of paths that will be used after as the kernel size of the CNN.
To simplify the task, we started working in the real space $\mathbb{R}$ using BPSK modulated-samples. In this CNN, we need only to find the pseudo-inverse which will have the same order as the filter modelling the channel, that is why we need only one kernel that its size is the estimated number of paths. An input matrix of shape (K;1) is considered, where K corresponds to the number of samples.

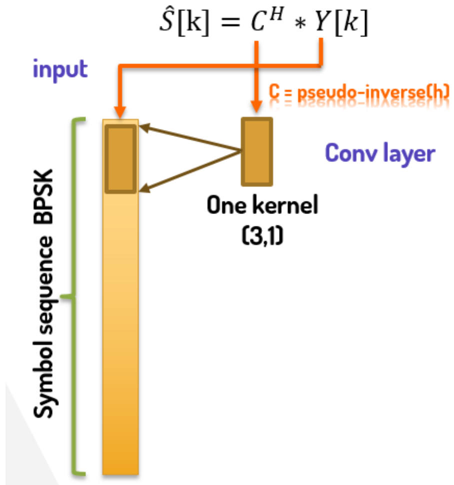


FIGURE 3.5: The CNN model for BPSK Modulation.

For an example of multi-path channel with 3 paths, we used the above configuration of the CNN (one layer, one kernel with size (3,1)), to equalize the BPSK samples. The kernel (Wiener filter) of the convolutional layer coefficients are adjusted using the MMSE loss function and the adam optimizer.
After the training stage using 10e6 BPSK samples for 1000 epochs and 20 as the batch-size, we were able to imitate the performance of the analytical method. As a measure of performance we used BER curve as it is shown below.

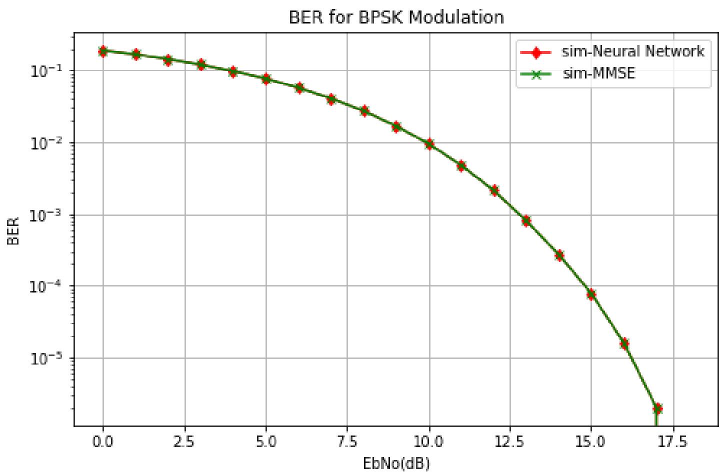


FIGURE 3.6: BER for the equalization of BPSK Modulation

The results of the equalization using CNN confirms the fact that the neural network can replace the conventional solution.
It is clear now that the CNN is able to equalize the samples in the case of the BPSK modulation. Thus, we can move forward and check the performance of this neural network on the 16-QAM modulated samples. In this case the samples are in the complex space $\mathbb{C}$, so, we need to split the input data into two channels; real and imaginary parts. The input matrix is a matrix of shape (K,2), where K corresponds to the number of samples and 2 introduces the number of the channel. The same

kernel will be applied for both channel (kernel-size(1,3)).

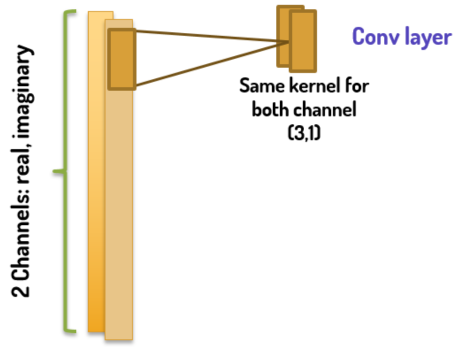


FIGURE 3.7: The CNN model for 16-QAM Modulation.

To visualise the performance of the CNN solution on the received 16-QAM samples with impairments, a plot in the IQ space was needed. The figure below shows that the CNN was able to equalize the transmit 16-QAM samples through a 3-path channel.

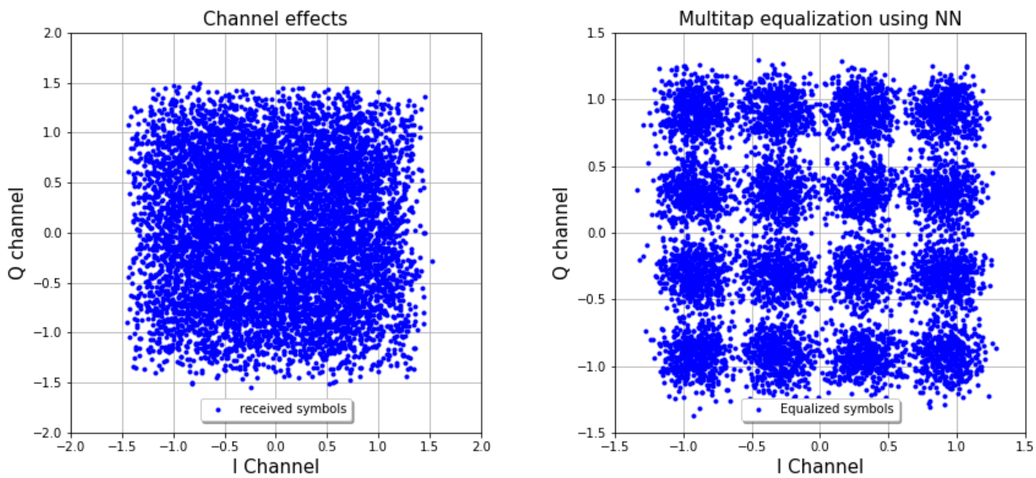


FIGURE 3.8: Left figure: Received 16-QAM samples with impairments - Right figure: Samples equalized by the CNN.

The BER curve was used to measure the performance of CNN method and to compare it to the analytical method which is based on product of the crosscorrelation matrix of YX and the auto-correlation matrix of Y as it is mentioned in the section 2.4.2 .

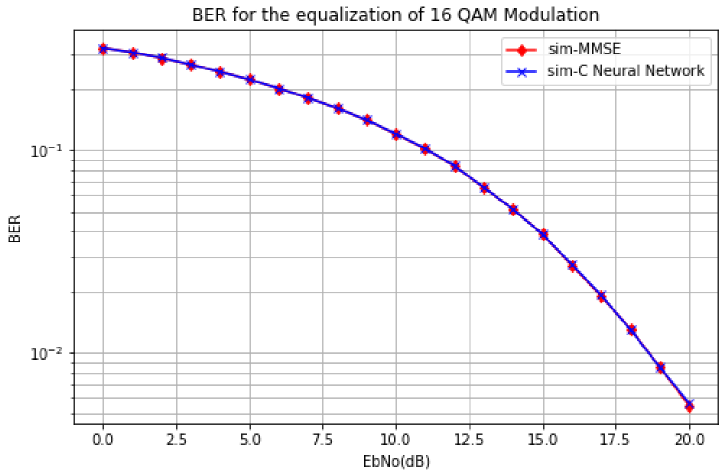


FIGURE 3.9: BER for the equalization of 16-QAM Modulation.

The results in the figure above confirms that the CNN method is efficient to equalize the 16-QAM samples. In fact, the proposed CNN model reaches theoretical optimal BER performance over 3-path channel.

3.3.2 Nonlinear method:Decision Feedback Equalizer(DFE)

Conventional Solution

The Decision Feedback Equalizer(DFE) is one of the adaptive equalizer. The adaptation of the DFE is divided into a training stage and a tracking stage. A known data sequence, which it is known by training sequence, is used in the training stage to converge the tap gains to the optimum values rapidly. For the second stage, LMS algorithm is used to converge to the true coefficients and have the benefit of tracking the changes in the channel impulse response.

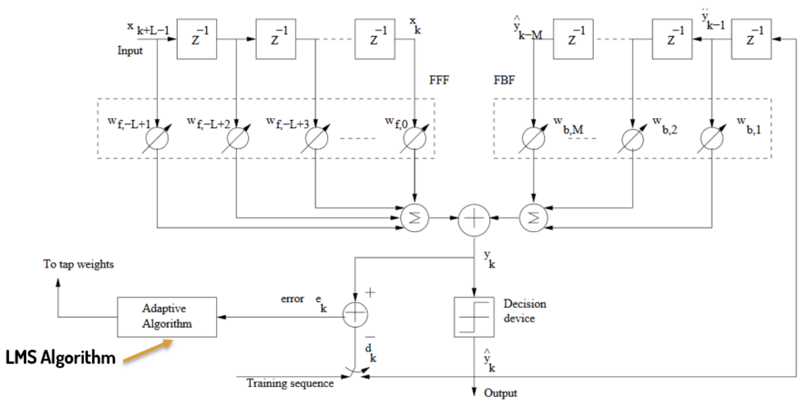


FIGURE 3.10: Algorithm of DFE.

The input vector of the feedforward filter FF and the feedback filter FB input are given by:

$$v_k = [x_{k+l-1} x_{k+l-2} ... x_k]^T \tag{3.8}$$

$$u_k = [\hat{y}_{k-1}\hat{y}_{k-2}...\hat{y}_{k-M}]^T \tag{3.9}$$

The equalized signal is given by equation, is the sum of outputs of the feedforward and feedback parts of the equalizer.

$$y_k = \sum_{l=-L+1}^{0} \omega_{f,l} x_{k-l} + \sum_{m=1}^{M} \omega_{b,l} \hat{y}_{k-l} \tag{3.10}$$

The feedforward filter and feedback filter coefficients are adjusted simultaneously to minimize the mean square error between the desired signal and the filter output. Using this algorithm concept, we were able to implement the DFE equalizer using python which has the some performance as the Matlab function.

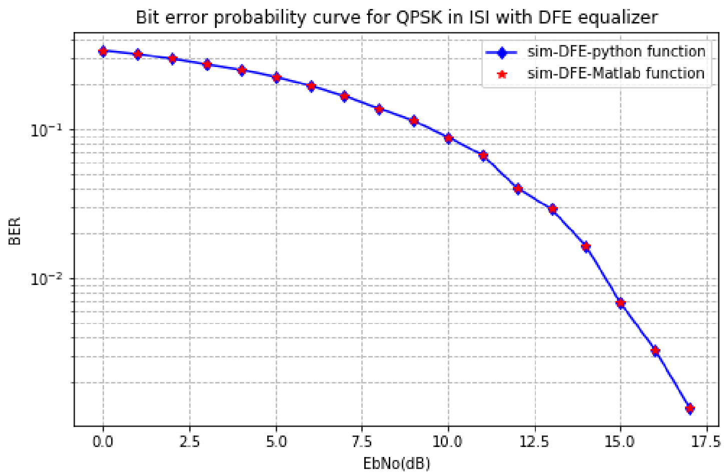


FIGURE 3.11: Bit error probability curve for QPSK in ISI with DFE equalizer.

NN Solution

The idea in this part is to find a NN architecture that can imitate the DFE function and it will be able to equalize the samples. The recurrent neural network (RNN) can be the kind of NN that can meet our demand. This kind of NN has a memory that allows learning across sequences of inputs rather than individual patterns. Unfortunately, we didn't have the opportunity to go further on this subject by lack of time.

3.4 Conclusion

We have proposed a new vision of using NN to equalize the channel in different configuration. The results confirms that the ANNs are able to imitate the analytical equalizer with low complexity. From this part we proposed a simple models of NN that will throw light on the *Black-box* issue of DNN.

Chapter 4

Conclusion

The goal of this project has been to investigate the possibility of replacing the conventional channel equalizer by a neural networks. A number of methods have been evaluated, both Linear(Wiener filter) and non-Linear (DFE) using dense and convolutional networks, which showed the new vision of using ANNs as a powerful and straightforward way of channel equalization.
In fact this work showed that a low complex neural network was enough to reduce and even remove the distortions of the one-path and mult-path channels. A simple CNN with one layer and 2 filter (6 parameters) was able to equalize the one-path channel and gives the same results as the conventional solution.
The multi-path channel presented more difficulties because of Inter-Symbol Interference. After the training stage the CNN was able to equalize the samples and have the same result as the analytical method.
This work demonstrates that we don't need a *deep* neural network and a lot of data to achieve a high performances, on the contrary, a low complex neural networks were enough and efficient especially for IOT application because its dedicated hardware are widely available with low cost and low-energy consumption. On the other hand, this does not prevent that in some situation DNN can be more interesting. Further improvements may be achieved by using recurrent neural network (RNN) to create a NN version of the decision feedback equalizer (DFE).

Appendix A

NN models

A.1 NN model for one-path channel

```
import tensorflow as tf

def one_tap_model(m):
    tf.keras.backend.clear_session()
    model = Sequential([Dense(2 , input_shape=(2,)),])
    model.compile(loss='mean_absolute_error',
    optimizer='adam', metrics=['accuracy'])
    return(model)
```

LISTING A.1: One-tap equalization using densNN

A.2 NN model for Multi-path channel: Wiener filter

A.2.1 For BPSK samples

```

def BPSK_model(m):
    tf.keras.backend.clear_session()
    model = tf.keras.models.Sequential([
        tf.keras.layers.Conv1D(
            input_shape=(m,1),
            filters=1,
            kernel_size=3,
            padding='valid',
            activation='linear'),
         tf.keras.layers.Flatten(),
    ])
    model.compile(optimizer='adam', loss='mean_squared_error',metrics
    =[])

    return(model)
```

LISTING A.2: Wiener filter equalization using CNN (for BPSK)

A.2.2 For 16-QAM samples

```

def 16_QAM_model(m):
    tf.keras.backend.clear_session()
    model = tf.keras.models.Sequential([
        tf.keras.layers.Conv2D(
            input_shape=(m,1,2),
            filters=2,
            kernel_size=(3,1),
            padding='valid',
            activation='linear'
```

```
                ),

        ])
        model.compile(optimizer='sgd', loss='mean_squared_error',metrics
       =[])

        return(model)
```

LISTING A.3: Wiener filter equalization using CNN (for 16-QAM)

Bibliography

[1] J.Gaeddert A.Fehske and J.H.Reed. "A new approach to signal classification using spectral correlation and neural networks ". In: *IEEE Int. Symp. New Frontiers in Dynamic Spectrum Access Networks* (2005), pp. 144–150.

[2] Jain L. (eds) Bianchini M. Maggini M. *Handbook on Neural Information Processing*. Intelligent Systems Reference Library. Springer, 2013.

[3] E.E.Azzouz and A.K.Nand. "Modulation recognition using artifi-cial neural networks ". In: *in Proc. Automatic Modulation Recognition of Communication Signals* (1996), pp. 132–176.

[4] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[5] M.Ortuno I.Ortuno and J.Delgado. "Error correcting neural networks for channels with Gaussian noise". In: *in Proc.IJCNN International Joint Conference on Neural Networks* 4 (1992), pp. 295–300.

[6] J.Cid-Sueiro and A.R.Figueiras-Vida. *Digital equalization using modular neural networks: an overview*. in Proc. Signal Processing in Telecommunications. Springer, 1996. ISBN: 978-3-540-76019-1.

[7] Q.Zhang J.Sjoberg and L.Ljung. "Nonlinear black-box modeling insystem identification: a unified overview". In: *Automatica* 31.12 (1995), pp. 1691–1724.

[8] Keras. *Dense Layer*. URL: https://www.tutorialspoint.com/keras/keras_dense_layer.htm. (accessed: 28.06.2020).

[9] S. B. Kotsiantis. "Supervised Machine Learning: A Review of Classification Techniques ". In: *Informatica* 31 (2007), pp. 249–268.

[10] K.Pulliyakode and S.Kalyani. *Reinforcement learning techniques for outer loop link adaptation in 4G/5G systems*. URL: https://arxiv.org/pdf/1708.00994.pdf. (accessed: 24.06.2020).

[11] Farhad E. Mahmood. *Mobile Radio Propagation Prediction for Two Different Districts in Mosul-City*. URL: https://www.intechopen.com/books/matlab-a-fundamental-tool-for-scientific-computing-and-engineering-applications-volume-2/mobile-radio-propagation-prediction-for-two-different-districts-in-mosul-city. (accessed: 28.06.2020).

[12] Piyush Malhotra. *Convolution and Variants*. URL: https://predictiveprogrammer.com/convolution-and-variants/. (accessed: 28.06.2020).

[13] Hani Mehrpouyan. *Channel Equalizer Design Based on Wiener Filter and Least Mean Square Algorithms*. URL: http://www.mehrpouyan.info. (accessed: 28.06.2020).

[14] J.Sombria M.Ibukahla and F. Castanie. "Neural networks for modeling nonlinear memoryless communication channels". In: *IEEE Trans. Commun* 45.7 (1997), pp. 768–771.

[15] Francisco Ordaz-Salazar Modar Shbat and Javier Salvador. *Nonlinear Channel Equalization Approach for Microwave Communication Systems*. URL: https://www.intechopen.com/books/microwave-systems-and-applications/nonlinear-channel-equalization-approach-for-microwave-communication-systems. (accessed: 28.06.2020).

[16] C. M. Caramanis R. C. Daniels and R. W. Heath. "Adaptation in con-volutionally coded MIMO-OFDM wireless systems through supervisedlearning and SNR orderin". In: *IEEE* 59.1 (2010), pp. 114–126. DOI: 10.1109/TVT.2009.2029693.

[17] G.Gibson S.Chen and C.Cowan. "Adaptive equalization offinite non-linear channels using multilayer perceptron". In: *Elsevier Signal processing* 20.2 (1990), pp. 107–119.

[18] Chao-Kai Wen Tianqi Wang. *Deep Learning for Wireless Physical Layer:Opportunities and Challenges*. URL: https://arxiv.org/abs/1710.05312. (accessed: 28.06.2020).

[19] Robert Tibshirani et Jerome Friedman Trevor Hastie. *The Elements of Statistical Learning Data Mining, Inference, and Prediction*. Springer Series in Statistics. Springer, 2009.

[20] L.Dong U.Challita and W.Saad. *Proactive resource management in LTE-U systems: A deep learning perspective*. URL: https://arxiv.org/pdf/1702.07031.pdf. (accessed: 24.06.2020).