Machine Learning-Assisted CRAN Design with Hybrid RF/FSO and Full-Duplex Self-Backhauling

by

Seyedrazieh Bayati

M.A.Sc., Amirkabir University of Technology, 2016

B.A.Sc., Sahand University of Technology, 2013

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Applied Science

in

THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

(Electrical and Computer Engineering)

The University of British Columbia

(Vancouver)

December 2020

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Machine Learning-Assisted CRAN Design with HybridRF/FSO and Full-Duplex Self-Backhauling

submitted by **Seyedrazieh Bayati** in partial fulfillment of the requirements for the degree of **Master of Applied Science** in **Electrical and Computer Engineering**.

Examining Committee:

Lutz Lampe, Electrical Engineering, UBC
Supervisor

Cyril Leung, Electrical Engineering, UBC
Supervisory Committee Member

Abstract

The ever increasing demand for higher data rates, lower latency communication, and a more reliable mobile network has led us toward the 5th generation (5G) of mobile networks. In 5G, resource allocation is one of the most challenging problems. Conventionally, model-driven methods, and analytical approaches have been used to allocate resources optimally. Despite accuracy, these methods often result in a non-convex optimization problem that is inherently challenging to handle and require proper convex approximation. To overcome such drawbacks, we need more efficient resource allocation techniques in the 5G mobile network.

This research will study the downlink of a cloud radio access network. The cloud radio access network enables coordinated beamforming and better interference management in ultra-dense networks. This architecture's bottleneck is backhaul capacity restriction limiting the benefits that the cloud radio access network offers. We will use hybrid radio frequency and free-space optical links to address the backhaul capacity limitation. Also, to improve the throughput and increase the spectral efficiency of the radio-frequency links, we propose in-band full-duplex self-backhauling radio units. After formulating the mathematical model and solving it with analytical approaches, we will introduce a novel solution for the proposed scenario and show that it outperforms the state-of-the-art half-duplex backhaul technology provided enough self-interference cancellation under various weather conditions.

We will derive a joint optimization problem to design the backhaul and access link precoders and quantizers subject to the fronthaul capacity, zero-forcing, and power constraints. We will show that this problem is non-convex and computationally intractable and approximate it with a semi-definite programming that can be

effectively solved by alternating convex optimization. We also employed Compute Canada computational resources for solving mentioned semi-definite programming. The computational complexity of the proposed optimization approach motivates us to employ machine-learning-based optimization methods that recently received much recognition in academia and industry. We use supervised and unsupervised deep neural networks for learning the optimal resource allocation strategy and achieved 80% of the performance compared to the proposed analytical approach with only a fraction of computational cost. To meet all feasibility constraints of the problem, we also propose customized activation functions and post-processing steps.

Lay Summary

Resources in the mobile communication networks, such as power and frequency, must be optimally managed and allocated to devices in order to achieve a good quality of service. On the one hand, the increase in the number of connected devices and the current growing demand for fast Internet connection has made the resource allocation problem one of the focus points in the wireless communication systems. On the other hand, the advent of the 5th generation (5G) of mobile network technology also has strictly imposed constraints on network resources and calls attention to the importance of efficient resource allocation in 5G. Achieving effective mobile communication in ultra-dense metropolitan areas and providing a good quality of service for mobile users with different demands requires a very efficient resource allocation technique. This thesis investigates the state-of-the-art technologies that enable fast and reliable wireless communication and proposes a novel network design for 5G. We introduce a machine learning-based resource allocation approach that reduces the computational cost of solving resource allocation problems compared to the conventional analytical methods.

Preface

This dissertation is original, unpublished, independent work by the author Seyedrazieh Bayati.

Table of Contents

List of Tables

List of Figures

Glossary

4G	4th Generation
5G	5th Generation
ANN	Artificial Neural Network
BS	Base Stations
CNN	Convolutional Neural Network
COMP	Coordinated Multi-Point
CP	Central Processor
CRAN	Cloud Radio Access Network
CSI	Channel State Information
DNN	Deep Neural Network
EMBB	Enhanced Mobile Broadband
FD	Full Duplex
FSO	Free-Space Optical
HD	Half Duplex
IBFD	In-Band Full-Duplex
LED	Light-Emitting Diode
MAE	Mean Average Error
ML	Machine Learning
MMTC	Massive Machine-Type Communications

MSE Mean Square Error

OOK On-Off Keying

PSD Positive Semi-Definite

RAN Radio Access Network

RF Radio Frequency

RU Radio Unit

SE Spectral Efficiency

SI Self-Interference

SIC Self-Interference Cancellation

SVD Singular Value Decomposition

URLLC Ultra-Reliable Low-Latency Communications

ZF Zero-Forcing

Acknowledgments

I would first like to thank my supervisor, Professor Lutz Lampe, whose expertise was invaluable in developing the study. His insightful feedback encouraged me to push my boundaries and think sharp.

I want to thank my parents, Seyedjafar Bayati and Mahnaz Hajian, and my beloved siblings, Seyedmohsen and Seyedmaryam for always being there for me. They always encouraged me to be the best me, never defined boundaries for my dreams, and supported me, no matter what.

Finally, I could not have completed this dissertation without the loving support of my best friend and companion, Faramarz Jabbarvaziri, who provided emotional comfort as well as pleasant distractions to rest my mind outside of my research.

Chapter 1

Introduction

1.1 Background and Motivations

Nowadays, the importance of a reliable, high-speed mobile network is not hidden from anyone. Especially during the pandemic, the world witnessed the importance of a reliable mobile network and used 4th Generation (4G) capabilities to cope with the new situation. Every aspect of human life, such as education, work, and healthcare, changed dramatically, and social distancing became possible via online connection and mobile connectivity. Based on [5], in 2025, the 5th Generation (5G) of mobile networks will support more than 2.8 billion subscriptions. Simultaneously, with an increase in data rate requests per user, due to the advances in augmented reality, online gaming, and high-quality videos, we expect a considerable increase in data rate demands in 5G.

5G cellular networks are supposed to support the three main objectives of massive machine-type communications Massive Machine-Type Communications (MMTC), Ultra-Reliable Low-Latency Communications (URLLC), and Enhanced Mobile Broadband (EMBB). Given these requirements and with the enormous increase in the number of devices, data rate demand, and diversity of connected users (from low-throughput, low-power sensors and wearable devices to high-throughput services such as online gaming), 5G cellular networks should be designed to be more efficient. It means 5G networks need a more structured and optimized way of managing resources such as spectrum, capacity, and energy. The Cloud Radio

Access Network (CRAN) concept helps to manage resources more efficiently by providing an intelligent structure to apply the state-of-the-art techniques [6, 7].

To be more specific, considering the limited frequency bands that are already over-utilized, the spectrum's more efficient use is crucial to accommodate this rapidly growing demand. One way to improve mobile networks' spectral efficiency is through the coordination of base stations to minimize inter-cell interference. Coordinated beamforming is an effective way to preclude base stations or Radio Unit (RU)s from causing interference in their neighbors' coverage areas, and CRAN is a new technology that is proposed to realize this goal. To do so, however, high-throughput backhaul connections are required to link the RUs to a Central Processor (CP) unit. The data of all users is needed to be shared with each individual RU or base station, and it requires that the backhaul capacity be multiple times larger than the capacity of uncoordinated RUs. To provide a high-enough backhaul capacity, the application of optical connections is considered for the backhaul. Notwithstanding the outstanding performance of the optical fiber solutions in providing a high-throughput backhaul link, the massive cost of deployment and maintenance has hindered the industry from utilizing the benefits of coordinated transceiver nodes.

In recent years, Free-Space Optical (FSO) communication is suggested as a cost-efficient solution for accomplishing the requirements of the backhaul of CRAN. In FSO, the optical signal generated by a laser is transmitted through the air, and an optical detector at the receiver converts it to electrical signals. However, FSO is highly sensitive to weather conditions and air quality as an optical signal can be easily obscured by fog, rain, and dust. Therefore, an RF link is needed as an auxiliary means of communication in poor weather conditions. Such a system is called hybrid RF/FSO backhaul, and it is one of the most promising methods for commercialization of CRAN in 5G [8].

1.1.1 Cloud Radio Access Network

Centralizing and virtualizing the conventional Radio Access Network (RAN) offers dynamic node cooperation ability, which results in coordinated connectivity across base stations Base Stations (BS) and better resource allocation. The CRAN shown

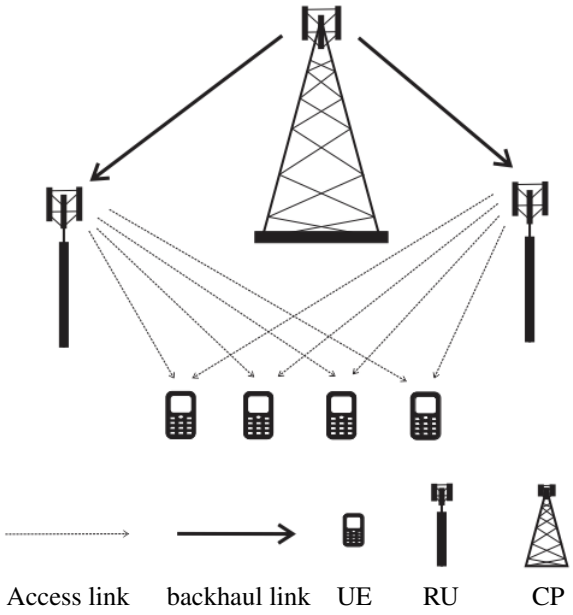

Figure 1.1: Illustration of the downlink CRAN system model. The CP is connected to two RUs with backhaul links and RUs serve four users cooperatively via access links.

in Figure 1.1, is a promising network architecture that realizes centralization and virtualization in the mobile network and can play an essential role in accomplishing 5G demands. Also, coordinated connectivity facilitates inter-cell interference management in the network. This can make the deployment of dense networks feasible and consequently increases spectral efficiency [6, 9, 10].

One of the main bottlenecks of CRAN is the limited backhaul capacity that connects the CP to the RU. This limitation can mask the CRAN's main advantages—coordinated beamforming, efficient resource allocation, and interference mitigation capability [11].

Multiple data transmission strategies can be applied in CRANs to deal with the limited capacity, including the data-sharing method, compression-based method, and a hybrid of them [6, 12–14]. In the data sharing method, each user is assigned to a cluster of RUs, and it's data is shared with all RUs in that cluster. Then, each

user is served by all RUs of the cluster via a joint beamforming scheme, which is known as the Coordinated Multi-Point (COMP) technique [12]. The bigger the cluster is, the more cooperation among RUs, and higher spectral efficiency can be achieved. In a compression-based scheme, the capacity restriction is handled through data quantization, which leads to quantization noise [13]. The backhaul capacity specifies the level to which the data needs to be quantized; therefore, a larger backhaul capacity results in lower quantization noise. According to [15], for medium to the high capacity backhaul links, which is the case for 5G, the compression-based approach outperforms the data-sharing method. Hence, providing a reliable and high-speed backhaul connection with acceptable quantization noise level is a plausible solution to deploy CRAN in a 5G network and meet its high spectral efficiency requirement.

1.1.2 5G Backhauling

Backhaul refers to the connection between the CP and RUs in the mobile network (see 1.2). The prevalence of devices operating in the RF band and the fact that the RF bandwidth is very limited in capacity and license makes the RF band a bottleneck for the CRAN; not to mention the interference as another limiting factor for reusing radio frequency bands. With the rapid growth of high-speed communication demands, Radio Frequency (RF) no more suffice, and new alternatives are required for 5G.

Recently the application of FSO communication in CRAN is suggested [11, 16, 17]. In particular, it is shown that a hybrid RF/FSO communication link can considerably boost the capacity thanks to the large bandwidth of FSO, while offering acceptable reliability owing to the robustness of connection in RF [8, 18]. In addition, self-backhauling defined as the use of the same RF band for both the access link and the backhaul is another way to comply with the limitation of bandwidth in RF [19, 20].

Finally, the combination of the FSO and an RF link with self-backhauling can provide a hybrid backhaul link with high enough capacity and reliability while complying with the limited RF bandwidth. In the next subsections, we elaborate on FSO and self-backhauling RF bands.

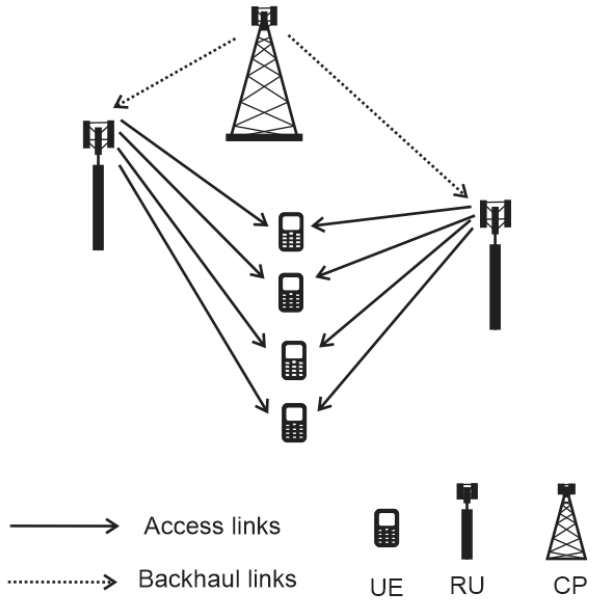

Figure 1.2: 5G CRAN.

1.1.3 Free Space Optical communication

Free-space optical communication is an alternative means that can overcome RF drawbacks, although it introduces new challenges. Unlike optical fiber, which is a high-speed communication link made from glass or plastic, FSO has much lower deployment cost. Easy implementation and low maintenance cost of FSO, comes with its less reliability and higher outage probability compared to optical fiber. Regardless of the current use cases of FSO, visible light has been one of the first communication tools of humanity used. Starting from Heliograph [1] and telegraph communication, visible light communication has been a non-commercial way of signaling from the early days. Nowadays, FSO is a transmission method in which laser and Light-Emitting Diode (LED)s are used to transmit data. Unlike fiber-optic communication, which is restricted to costly fibers made of glasses, FSO is

[1] A signaling device with sunlight

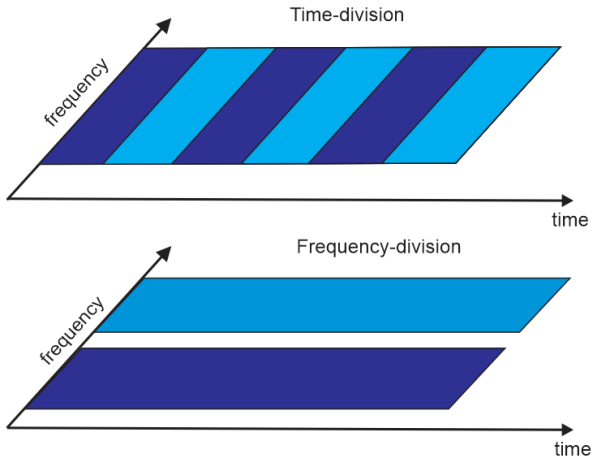

Figure 1.3: Time division and frequency division duplex technologies.

communicated through the air, thus has far less implementation and maintenance costs. Compared to RF communication, as FSO operates in higher bandwidths, it provides higher data rate—10 Gbps per wavelength [21]. More importantly, FSO is immune to interference. In clear weather conditions, FSO has less than 1dB/km attenuation, making it a perfect candidate for communication over high distances [21]. Finally, unlike the RF, FSO benefits from the availability of a license-free spectrum, which is an excellent means for cutting the deployment costs.

Despite these advantages, FSO has some concerning drawbacks. The performance of the FSO is susceptible to weather conditions, and it can experience from 5 up to 350 dB/km attenuation in rainy and foggy weather conditions, respectively [22]. Also, as the maximum possible power is limited by safety concerns, more effective signal processing is required to overcome the signal attenuation of FSO [23, 24].

1.1.4 RF Self-Backhauling

One way to deal with the limitation of RF bandwidth is frequency reuse. Traditionally, wireless nodes work in a half-duplex manner, meaning that the send and receive signals use the same band but different times or the same time but differ-

ent frequency bands (see Figure 1.3). Using the same frequency band for both the transmit and receive simultaneously is referred to as In-Band Full-Duplex (IBFD). Also, as mentioned, using the same frequency for backhaul and access link at the same time, is called in-band full-duplex self-backhauling. The main limiting factor in IBFD technology is the interference of the transmit signal that masks the received signal. Lack of practical interference cancellation techniques held back the industry to deploy IBFD in practical scenarios. As shown in Figure 1.4, there is always a Self-Interference (SI) signal from the transmitter to the receiver in both separate-antenna and shared-antenna systems [25]. To benefit from the IBFD, there must be very strong isolation between the transmit and receive signals. The challenge lies in the fact that the transmission power is usually much larger than the received power; thus even a small residual interference can mask the received signal.

Recent advancements in full-duplex RF communication have nearly doubled RF communication's Spectral Efficiency (SE), creating the opportunity to enhance the fronthal capacity. A combination of Self-Interference Cancellation (SIC) methods, e.g., propagation, analog, and digital SIC, provide sufficient isolation that near double the spectral efficiency can be achieved in [20, 26–31]. Also, it is shown that the Full Duplex (FD) communication scheme operates on a lower power budget compared to Half Duplex (HD) scheme [31].

The advancements above motivated us to incorporate hybrid RF/FSO schemes and IBFD RF self-backhauling techniques to realize a high-throughput and reliable backhaul for CRAN.

1.1.5 Complexity Crunch in CRAN

The concept of CRAN has a large degree of abstraction, meaning that it refers to multiple modules such as the data-compression module, CP-to-RU beamformer, and RU-to-user beamformer. Each of these modules' design problem, even individually, is indeed a time-consuming optimization task. Hence, optimal CRAN design with several nodes in a dense 5G network poses a significant challenge in terms of computational complexity.

However, the recent advancements of data-driven Machine Learning (ML) ap-

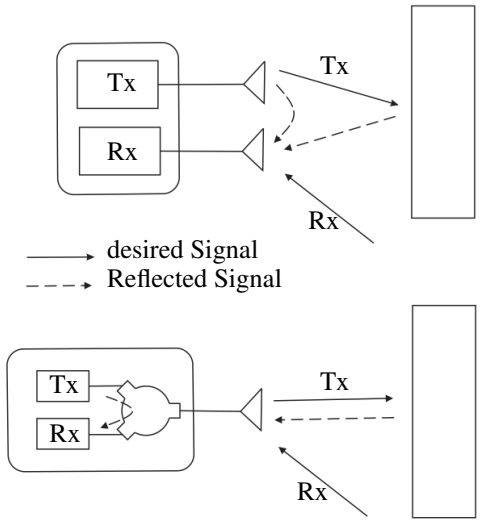

Figure 1.4: Illustration of self-interference concept in separate-antenna full-duplex and shared-antenna full-duplex systems.

proaches in addressing complicated design problems motivate us to view CRAN optimization in light of these powerful tools. Machine learning approaches, specifically Deep Neural Network (DNN), are shown to be able to generalize the patterns and relations hidden in the training data to the new test case enabling researchers to apply them in many different design problems successfully.

In this study, we are interested in the great potential of DNN to learn to solve optimization problems. For instance, [32], [33] propose DNNs that learn how to solve resource allocation problems. It has been shown that these methods cost less time as once trained, a DNN can produce the optimal results with low complexity. This motivates us to study the application of DNNs in 5G CRAN design.

1.2 Objectives and Contributions

It is shown in [15] that CRAN performance can be profoundly affected by the lack of high backhaul capacity. This obstacle can be explained by the added quantization noise, which affects the received signal's quality at the user's side.

Mostafa et al. in [8] employed a HD time-sharing scheme to realize RF self-backhauling in a hybrid RF/FSO backhaul. We also explored a hybrid RF/FSO with FD self-backhauling in a CRAN architecture to address RF spectrum limitations, and unlike [8], we will transmit RF signals at the same time without employing time-sharing. In HD, the backhaul RF capacity is shown to be much more restricted than FD self-backhauling [34]. Therefore, the compression-based approach performs much better when coupled with FD communication.

In this project, we aim at maximizing the downlink user sum-rate under transmit power constraints at both the CP and RUs, a Zero-Forcing (ZF) requirement for eliminating the multi-RU interference, and the backhaul capacity limitation.

We will adopt a hybrid RF/FSO with IBFD RF link for the backhaul of CRAN and consider the joint optimization of backhaul RF beamforming, data-compression, and access link beamforming to maximize the user sum-rate. We will analytically derive a closed-form formula for the sum rate and form a non-convex optimization problem. We propose a transformation to simplify this problem to a weighted sum mean-square error (MSE) minimization, which is a semi-definite programming, and adopt an alternating optimization approach to solve it and show its superiority over the existing hybrid RF/FSO with HD self-backhauling [8].

Moreover, we will illustrate that CRAN optimization's computational complexity is quite large and adopt data-driven approaches to overcome this challenge. In particular, we apply deep neural networks in both supervised and unsupervised manners to transfer much of the computational complexity to the offline training phase. This way, we achieve a CRAN design methodology that outperforms both the state-of-the-art method proposed by [8] and our own proposed conventional optimization in terms of user-sum rate and computational complexity.

1.3 Organization

In Chapter 2 the definition of the scenario, system model, and formulation of the problem are presented. In Chapter 3, we propose a semi-definite programming solution for CRAN downlink optimization and achieve superior results compared to the state-of-the-art methods. In Chapters 4 and 5 we present a supervised and an unsupervised machine learning solution, respectively, to reduce the computational

complexity of the optimization task. Finally, the Chapter 6 is dedicated to summary and conclusions.

Chapter 2

Problem Formulation

2.1 System Model

We consider a CRAN network composed of one CP, M RUs, and K users. The CP is assumed to apply the compression-based method [13] to transmit K independent data streams to users via M RUs, as illustrated in Figure 2.1. The backhaul links connect the CP to the RUs with a multi-input multi-output (MIMO) RF link with bandwidth B^{RF} and one FSO link with bandwidth B^{FSO}. The number of antennas for downlink transmission at the CP is denoted N^{CP}, and the number of transmit and receive antennas at the RU is N^{RU}. It is assumed that the CP has access to the complete channel state information of both the backhaul and access links. Each RU acts as an FD relay between the CP and users, and thus, both the access and backhaul links transmit on the same frequency band simultaneously. The RUs linearly precode and jointly transmit the data symbols intended for each user.

2.1.1 Hybrid RF/FSO fornthaul channel model

In Chapter 1, we discussed the importance of fornthaul link reliability and capacity in CRAN architecture and 5G. fornthaul link capacity is the most important limiting factor on cooperative beamforming performance and resource allocation performance in CRAN. Providing a reliable high data rate backhaul link can significantly affect 5G characteristics and is vital to make use of CRAN advantages over RAN structure. In our model, we consider hybrid RF/FSO links in backhaul to

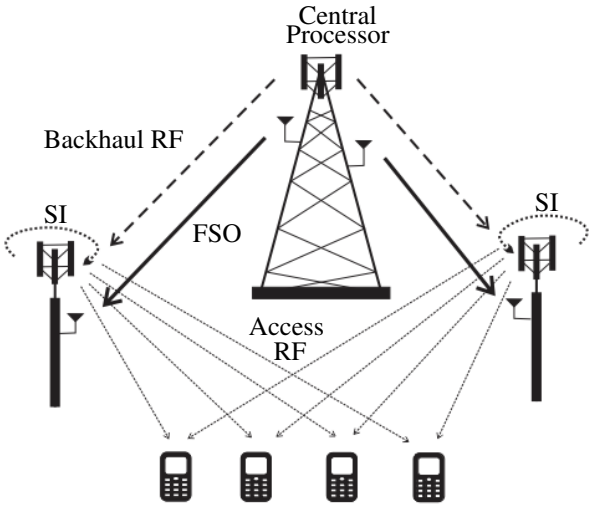

Figure 2.1: Illustration of the downlink CRAN. The CP is connected to two self-backhauling RUs with a hybrid RF/FSO backhaul. The RUs serve four users cooperatively via IBFD RF access links.

address this vital need. As mentioned before, RF and FSO can act complementary because of different carrier frequency utilization. High frequency in FSO provides high data rate though very dependent on weather condition, and lower frequency bands in RF link provides lower but reliable data rate for backhaul link.

2.1.2 Full Duplex Radios

For many years, in-band full-duplex communication seemed impossible. As Andrea Goldsmith once contended, "It is generally not possible for radios to receive and transmit on the same frequency band because of the interference that results." [35].

However, recent advancements in designing and manufacturing transceivers that provide superb analog isolation together with the state-of-the-art digital cancellation methods that almost wholly cancel the remaining interference, made it possible to deploy IBFD communication links in the RF band. Bharadia, D et al. proposed the first design and implementation of an IBFD system for WiFi 802.11ac [3]. They propose two main modules to render the self-interference almost negligible —1) an analog transceiver circuit that provides 60dB isolation between the

transmit and receive signal and 2) a digital cancellation method that attenuates the remaining interference by an additional 50dB. Achieving 110dB isolation makes it possible to render the self-interference under the noise floor or very close to the noise floor. Reaching the noise floor practically means that the self-interference signal is fully canceled as it is not distinguishable from the receiver's thermal noise.

In practice, the transmit signal has multiple components—the main signal, harmonics caused by hardware nonlinearities, and the transmitter noise. Each of these signals requires different cancellation techniques and different attenuation levels to reach the noise floor. For example, in Figure 2.2, we can see the isolation requirements for a practical WiFi system. The main signal is 20dB, which requires 110dB cancellation to reach the noise floor. The signal harmonics are usually around -10dB, so 80dB cancellation is enough for them, and the transmitter noise is around -40dB, which requires only 50dB cancellation. [3] proposes cancellation methods for each of these components, and in their experimental results, they achieve near double the spectral efficiency of a conventional WiFi system.

In this work, we adopt the IBFD system proposed in [3] for the RF backhaul link; however, we consider and model some residual self-interference to obtain a more robust design. In Section 2.2.1, we provide more details on the IBFD self-interference model.

2.1.3 FSO Channel Model

The most practical modulation considered for modeling FSO channel is On-Off Keying (OOK) modulation, which works by intensity modulation of transmit x_m^{fso} and received y_m^{fso} diodes intensity, where $x_m^{fso} \in \{0, 2P_{fso}\}$ with average transmit power of P_{fso}. We can model FSO received intensity in the m-th RU with

$$y_m^{fso} = h_m^{fso} x_m^{fso} + n_m, \tag{2.1}$$

where FSO channel of the m-th RU h_m^{fso} modeled with considering three attenuation types, atmospheric, scintillation and geometric losses [36]. According to [8], with ϕ as the divergence angle of FSO, and r as the radius of optical opening, the FSO channel gain is $h^{fso} = h_l h_s h_g R$, where $h_l = e^{\sigma_d d_{\mathfrak{fh}}}$ is the atmospheric loss model, $h_s = \mathrm{GG}(a,b)$ is the Gamma-Gamma distribution of scintillation and

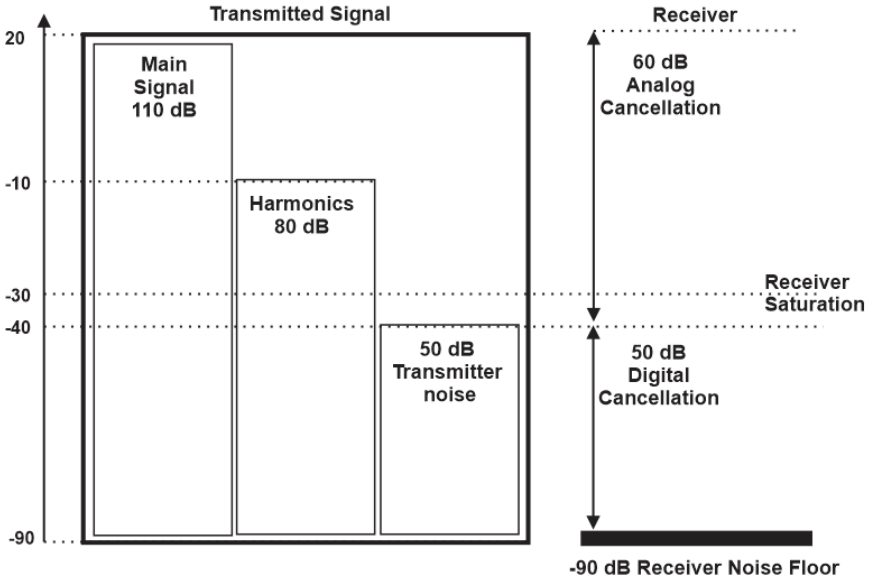

Figure 2.2: Illustration of noise floor and digital/analog cancellations [3]

$h_g = [\text{erf}(\sqrt{\pi}r/\sqrt{2}d_{fh}\phi)]^2$, in which erf(.) is the error function.

2.1.4 RF Backhaul Channel Model

For modeling backhaul link, we used path loss and antenna gain pattern (h_l) and Rician fading channel (h_s) to model line-of sight (LoS) propagation using link budget model. The channel gain is modeled as $h = h_l h_s$ with 5dB Rician fading factor indicating the ratio between the direct and scattered path powers, and line-of-sight path loss exponent of $n_{LoS} = 2.5$. Also, for all antennas, the antenna gain is assumed to be 3 dB. Link budget calculation is adopted form [37] and channel information and the simulation setup are provided in Table 3.2.

2.1.5 RF Access Channel Model

Similarly for downlink link we model channel gain as $h = h_l h_f$ where h_l is path loss and h_f is Rayleigh distribution fading gain. The distance between RUs and UEs assumed to be 10 meters fix and each RU's power budget is set to 23 dB. We calculate path loss with exponent of $n_{LoS} = 3.5$ in downlink. The system parameters

are summarized in Table 3.2, with the channel specifications and link models as in [8, 37].

2.2 Problem Formulation

Let $\mathbf{s} = [s_1, \ldots, s_K]^T$ be the vector of normalized Gaussian data symbols intended for users $\{1, \ldots, K\}$, $\mathbf{W}_m \in \mathbb{C}^{N^{\mathrm{RU}} \times K}$ denote the precoding matrix of the m-th RU toward users and $\mathbf{q}_m \sim \mathscr{CN}(0, \mathbf{Q}_m)$ denote the quantization noise with covariance matrix $\mathbf{Q}_m$ at the m-th RU due to data compression. The transmitted signal from the m-th RU to the users is denoted by $\mathbf{x}_m \in \mathbb{C}^{N^{\mathrm{RU}}}$ and can be formulated as

$$\mathbf{x}_m = \mathbf{W}_m \mathbf{s} + \mathbf{q}_m. \tag{2.2}$$

In the following sections we will introduce the constraints of our problem in detail.

2.2.1 Backhaul Capacity Constraint

In this subsection, we will compute the minimum rate required for the backhaul links. Assume that the RU is equipped with an IBFD transceiver, which receives the signals from the CP and transmits simultaneously to users in the same RF band. We can formulate the RF received signal at the m-th RU as

$$\mathbf{y}_m^{\mathrm{rf}} = \mathbf{G}_m^{\mathrm{H}} \mathbf{x}_{\mathrm{cp}} + \mathbf{I}_m^{\mathrm{si}} + \mathbf{n}_m, \tag{2.3}$$

where $\mathbf{n}_m \sim \mathscr{CN}(0, \sigma_n^2 \mathbf{I})$ is the additive white Gaussian noise, $\mathbf{I}_{\mathrm{si}}$ is the residual self-interference, and $\mathbf{x}_{\mathrm{cp}} \in \mathbb{C}^{N^{\mathrm{CP}}}$ is the transmitted signal by the CP. Also, $\mathbf{G}_m \in \mathbb{C}^{N^{\mathrm{CP}} \times N^{\mathrm{RU}}}$ is flat-fading MIMO backhaul RF channel. The self-interference signal is proportional to the transmitted signal, and therefore, the residual self-interference power at the m-th RU is proportional RU's transmit power. That is,

$$\mathrm{E}\{\mathbf{I}_m^{\mathrm{si}} (\mathbf{I}_m^{\mathrm{si}})^{\mathrm{H}}\} = \alpha \mathbf{P}_m^{\mathrm{si}}, \tag{2.4}$$

where $\mathbf{P}_m^{\mathrm{si}} = \mathbf{W}_m \mathbf{W}_m^{\mathrm{H}} + \mathbf{Q}_m$. We define α as a unit-less constant and it depends on RU's capacity in suppressing the self-interference [34, 38]. Hypothetical ideal

SIC makes each transmission node mutually orthogonal and results in $\alpha \mathbf{P}_m = 0$, while practical IBFD results in residual self-interference [39, 40].

The signal components intended for different RUs are separated by precoding at the CP. For this we assume that there is enough distance between the RUs to have a full rank MIMO channel and that the number of antennas at the CP is more than the total number of RU antennas, i.e., $N^{\mathrm{CP}} \geq MN^{\mathrm{RU}}$. Then, we can neglect interference between the signals for different RUs by imposing the zero-forcing constraint [41]

$$\mathbf{G}_j^{\mathrm{H}} \mathbf{V}_m \mathbf{G}_j = 0, \quad j \neq m, \tag{2.5}$$

where $\mathbf{V}_m$ is the covariance matrix of the transmitted signal intended for the m-th RU. The corresponding maximum communication rate in the RF link between the CP and the m-th RU can be approximated in the high signal-to-noise ratio (SNR) regime as

$$\begin{aligned} & C_m^{\mathrm{RF}}(\mathbf{V}_m, \mathbf{Q}_m, \mathbf{W}_m) \\ & = \ln \det \left(\mathrm{I} + (\alpha \mathbf{P}_m^{\mathrm{si}} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{G}_m^{\mathrm{H}} \mathbf{V}_m \mathbf{G}_m \right) \\ & \approx \ln \left(\det \left(\alpha \mathbf{P}_m^{\mathrm{si}} + \sigma_n^2 \mathbf{I} \right)^{-1} \det (\mathbf{G}_m^{\mathrm{H}} \mathbf{V}^m \mathbf{G}_m) \right) \\ & = \ln \det (\mathbf{G}_m^{\mathrm{H}} \mathbf{V}^m \mathbf{G}_m) - \ln \det \left(\alpha \mathbf{P}_m^{\mathrm{si}} + \sigma_n^2 \mathbf{I} \right), \end{aligned} \tag{2.6}$$

and the data rate is $B^{\mathrm{RF}} C_m^{\mathrm{RF}}$.

For the FSO links, it is assumed that commonly used on-off keying (OOK) modulation is applied [36]. The CP is assumed to be equipped with M OOK transmitters, and each RU has one optical receiver. For a given B^{FSO} as FSO signaling rate, channel gain h_{FSO}, optical power budget P_{FSO}, and thermal noise and background illumination σ_{FSO}^2, the maximum communication rate under OOK modulation can be calculated as [8]

$$C_m^{\mathrm{FSO}} = - \int_{-\infty}^{\infty} p(y) \ln p(y) dy - \ln (2\pi e \sigma_{\mathrm{FSO}}^2), \tag{2.7}$$

where

$$p(y) = \frac{1}{2\sqrt{2\pi\sigma^2}} (e^{\frac{-y^2}{2\sigma_{\mathrm{FSO}}^2}} + e^{\frac{-(y-2h_m^{\mathrm{FSO}} P_{\mathrm{FSO}})^2}{2\sigma_{\mathrm{FSO}}^2}}). \tag{2.8}$$

The corresponding data rate in the m-th FSO link is equal to $B^{\mathrm{FSO}}C_m^{\mathrm{FSO}}$. It is worth noting that due to the directional nature of FSO, they do not interfere with each other.

The backhaul link between the CP and the m-th RU needs to be able to support the transmission of the signal $\mathbf{x}_m$ in (2.2), which is the quantized version of the signal $\hat{\mathbf{x}}_m$ that would ideally be transmitted from the m-th RU in the absence of backhaul capacity limitations. The required communication rate to encode $\hat{\mathbf{x}}_m$ into $\mathbf{x}_m$ with covariance matrix $\mathbf{Q}_m$ of the quantization noise is given by [13]

$$I_m^{\mathrm{data}} = \ln \frac{\det(\mathbb{E}\{\hat{\mathbf{x}}_m \hat{\mathbf{x}}_m^{\mathrm{H}}\} + \mathbf{Q}_m)}{\det(\mathbf{Q}_m)}, \tag{2.9}$$

where $\hat{\mathbf{x}}_m$ is the uncompressed transmitted signal from the m-th RU in the absence of backhaul capacity condition. Assuming the same transmission bandwidth for the RF backhaul and user links, this leads to the capacity constraints

$$B^{\mathrm{RF}} R_m^{\mathrm{data}} \leq B^{\mathrm{FSO}} C_m^{\mathrm{FSO}} + B^{\mathrm{RF}} C_m^{\mathrm{RF}}, \quad \forall m \in M. \tag{2.10}$$

Thus, we have

$$\begin{aligned} & B^{\mathrm{RF}} \ln\det(\mathbf{W}_m \mathbf{W}_m^{\mathrm{H}} + \mathbf{Q}_m) - B^{\mathrm{RF}} \ln\det(\mathbf{Q}_m) \\ \leq & B^{\mathrm{FSO}} C_m^{\mathrm{FSO}} + B^{\mathrm{RF}} \ln\det(\mathbf{G}_m^{\mathrm{H}} \mathbf{V}^m \mathbf{G}_m) \\ & - B^{\mathrm{RF}} \ln\det(\alpha \mathbf{P}_m^{\mathrm{si}} + \sigma^2 \mathrm{I}), \end{aligned} \tag{2.11}$$

which is non-convex. In chapter 3, we transform this problem to a convex semi-definite problem and employ alternating convex optimization to solve it.

2.2.2 Power Constraints

The RF beamforming transmitters at both CP and RUs operate under power constraints that can be written as

$$\mathrm{Tr}\big(\mathbb{E}\{\mathbf{x}_{\mathrm{cp}} \mathbf{x}_{\mathrm{cp}}^{\mathrm{H}}\}\big) = \mathrm{Tr}\Big(\sum_{m=1}^{M} \mathbf{V}_m\Big) \leq P_{\mathrm{cp}} \tag{2.12}$$

$$\mathrm{Tr}(\mathbb{E}\{\mathbf{x}_m \mathbf{x}_m^{\mathrm{H}}\}) = \mathrm{Tr}(\mathbf{W}_m \mathbf{W}_m^{\mathrm{H}} + \mathbf{Q}_m) \leq P_m, \tag{2.13}$$

in which P_{cp} and P_m are the CP and RUs' power budgets, respectively.

2.2.3 Joint Optimization Problem

Given the setup described in the previous section, our objective is to maximize the user's sum-rate. While the type of objective and the capacity and power constraints by means of linear precoders and quantizers are alike those considered in the previous work [8], because of the FD self-backhauling, we are facing a different objective function and capacity constraints. In particular, our optimization problem is formulated as

$$\underset{\mathbf{V}_m,\mathbf{W}_m,\mathbf{Q}_m}{\text{maximize}} \quad R_{\text{sum}} \tag{2.14a}$$

$$\text{s.t.} \quad (2.5),(2.11),(2.12),(2.13) \quad \mathbf{V}_m \succcurlyeq 0, \mathbf{Q}_m \succcurlyeq 0, \tag{2.14b}$$

where, $(.) \succcurlyeq 0$ stands for positive semi-definiteness. The objective function is the weighted sum of maximum communication rate of users

$$\mathrm{R}_{\text{sum}} = B^{\text{RF}} \sum_{k=1}^{K} \gamma_k \log_2 \left(1 + \frac{|\mathbf{h}_k^{\mathrm{H}} \mathbf{w}_k|^2}{\sum_{l \neq k} |\mathbf{h}_k^{\mathrm{H}} \mathbf{w}_l|^2 + \mathbf{h}_k^{\mathrm{H}} \mathbf{Q} \mathbf{h}_k + \sigma_n^2} \right), \tag{2.15}$$

where

$$\mathbf{h}_k = [\mathbf{h}_{1,k}^{\mathrm{T}}, \ldots \mathbf{h}_{M,k}^{\mathrm{T}}]^{\mathrm{T}}, \ \mathbf{w}_k = [\mathbf{w}_{1,k}^{\mathrm{T}}, \ldots \mathbf{w}_{M,k}^{\mathrm{T}}]^{\mathrm{T}} \tag{2.16}$$

and $\mathbf{h}_{m,k}$ and $\mathbf{w}_{m,k}$ are channel gain and beamforming vectors from the m-th RU to the k-th user, respectively, and $\mathbf{Q} = \text{diag}(\mathbf{Q}_1, \ldots \mathbf{Q}_M)$. Also, γ_k is the weight of k-th user. This problem is non-convex and difficult to solve. In general, there is no guarantee for finding an optimal solution for this problem. However, to deal with the non-convexity, in the next chapter we first transform this problem into a weighted sum-mean-square error (MSE) minimization problem as suggested by [8, 42], and then approximate the capacity constraint by a convex subset. We will show in the next section that an alternating convex optimization can be applied to find a sub-optimal solution for this problem.

Chapter 3

CRAN Optimization

3.1 Introduction

In the previous chapter, we formulated the joint optimization problem of downlink CRAN with hybrid RF/FSO with IBFD RF. We focused on the joint design of access and backhaul precoders ($\mathbf{W}, \mathbf{V}$), and access link quantizer ($\mathbf{Q}$) in order to maximize the users sum-rate, considering power and ZF constraints and backhaul capacity limitation. This optimization problem is, however, non-convex. This chapter will show that this non-convex problem, which is hard to optimize, can be transformed into semi-definite programming and an alternating convex optimization algorithm can be used to solve it.

3.2 Transformation of the CRAN Optimization Problem

3.2.1 Transformation of the Objective Function

It has been shown in [42] that maximizing the non-convex sum-rate objective function is equivalent to a weighted sum-MSE minimization problem. By doing this transformation, (2.15) can be written as the following semi-definite objective

$$R' = \sum_{k=1}^{K} \gamma_k \beta_k \left(|g_k|^2 \left(\mathbf{h}_k^{\mathrm{H}} (\mathbf{W}\mathbf{W}^{\mathrm{H}} + \mathbf{Q}) \mathbf{h}_k \right) - 2\mathscr{R}e(g_k^* \mathbf{h}_k^{\mathrm{H}} \mathbf{w}_k) \right), \tag{3.1}$$

where

$$\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K]. \tag{3.2}$$

For a fixed set of optimization variables, we have

$$g_k = \frac{\mathrm{h}_k^{\mathrm{H}} \mathbf{w}_k}{\mathbf{h}_k^{\mathrm{H}} (\mathbf{W}\mathbf{W}^{\mathrm{H}} + \mathbf{Q})\mathrm{h}_k + \sigma^2}, \tag{3.3}$$

in which g_k is the scalar linear receive filter applied by k-th user and the corresponding optimal MSE weight is $\beta_k = 1/\mathrm{E}_k$, where $\mathrm{E}_k = \mathbb{E}\{|g_k^* y_k - s_k|^2\}$. This objective function is convex with respect to the individual optimization variables.

3.2.2 Convex Approximation of Capacity Constraint

Another source of non-convexity in the problem formulation is (2.11). Following [8], we deal with the non-concave function on the left-hand side of the inequality by employing conjugate function definition and Fenchel's inequality:

$$\begin{aligned} \ln\det(\mathbf{W}\mathbf{W}_m^{\mathrm{H}} + \mathbf{Q}_m)^{-1} \geq & -\mathrm{Tr}(\mathbf{Z}_m(\mathbf{W}\mathbf{W}_m^{\mathrm{H}} + \mathbf{Q}_m)) \\ & + \ln\det(\mathbf{Z}_m) + N^{\mathrm{RU}}, \end{aligned} \tag{3.4}$$

for some positive definite $N^{\mathrm{RU}} \times N^{\mathrm{RU}}$ matrix $\mathbf{Z}_m$. It has been shown in [8] that with $\mathbf{Z}_m = (\mathbf{W}_m\mathbf{W}_m^{\mathrm{H}} + \mathbf{Q}_m)^{-1}$, we can replace the non-convex inequality (2.11) with

$$\begin{aligned} & -\mathrm{Tr}(\mathbf{Z}_m(\mathbf{W}\mathbf{W}_m^{\mathrm{H}} + \mathbf{Q}_m)) + \ln\det(\mathbf{Z}_m) + N^{\mathrm{RU}} \geq \\ & -\frac{B^{\mathrm{FSO}}}{B^{\mathrm{RF}}} C_m^{\mathrm{FSO}} - \ln\det(\mathbf{G}_m^{\mathrm{H}} \mathbf{V}_m \mathbf{G}_{\mathrm{m}}) \\ & + \ln\det(\alpha\mathbf{P}_m^{\mathrm{si}} + \sigma_n^2\mathbf{I}) - \ln\det(\mathbf{Q}_m). \end{aligned} \tag{3.5}$$

The term $\ln\det(\alpha\mathbf{P}_m^{\mathrm{si}} + \sigma_n^2\mathbf{I})$ on the right hand side of (3.5) is non-convex. To deal with this term consider that the power of loop-back interference is less that or equal to the RU's transmit power, (i.e. $\mathrm{Tr}(\mathbf{W}_m\mathbf{W}_m^{\mathrm{H}} + \mathbf{Q}_m) \leq P_m$). For simplicity, we resort to the worst case scenario with the residual self-interference power being equal to αP_m. Thereby, we achieve a lower bound on the maximum achievable user's sum-

rate. According to [34] and [43], we assume that $\mathbf{I}_{\text{si}} \sim \mathscr{CN}(0, \sigma_{\text{si}}^2\mathbf{I})$, and thus

$$\text{Tr}(\sigma_{\text{si}}^2\mathbf{I}) = \alpha\text{Tr}(\mathbf{W}_m\mathbf{W}_m^{\text{H}} + \mathbf{Q}_m) \leq \alpha P_m \tag{3.6}$$

therefore, $\sigma_{\text{si}}^2 \leq \frac{\alpha P_m}{N^{\text{RU}}}$. So in the worst case scenario we will have

$$\alpha\mathbf{P}_m^{\text{si}} = \frac{\alpha P_m}{N^{\text{RU}}}\mathbf{I}. \tag{3.7}$$

Then we can write the inequality (3.5) as

$$\begin{aligned} &-\text{Tr}(\mathbf{Z}_m(\mathbf{W}\mathbf{W}_m^{\text{H}} + \mathbf{Q}_m)) + \ln\det(\mathbf{Z}_m) + N^{\text{RU}} \geq \\ &-\frac{B^{\text{FSO}}}{B^{\text{RF}}}C_m^{\text{FSO}} - \ln\det(\mathbf{G}_m^{\text{H}}\mathbf{V}_m\mathbf{G}_{\text{m}}) \\ &+\ln\det(\frac{\alpha P_m}{N^{\text{RU}}}\mathbf{I} + \sigma_n^2\mathbf{I}) - \ln\det(\mathbf{Q}_m). \end{aligned} \tag{3.8}$$

This inequality provides a convex set with respect to the design parameters, $\mathbf{W}_m, \mathbf{V}_m, \mathbf{Q}_m$, when $\mathbf{Z}_m$ is fixed and vice-versa. We can easily show that the inequality (3.8) is a subset of the non-convex inequality (3.5), therefore, it will not cause infeasible solutions.

3.3 Optimization Algorithm

For a given $\mathbf{Z}_m$ the following problem is a convex semi-definite programming and can be solved using alternating convex optimization [8]:

$$\underset{\mathbf{V}_m, \mathbf{W}_m, \mathbf{Q}_{\text{m}}}{\text{maximize}} \quad R' \tag{3.9a}$$

$$s.t. \quad (2.5), (2.12), (2.13), (3.8), \mathbf{V}_m \succcurlyeq 0, \mathbf{Q}_m \succcurlyeq 0 \tag{3.9b}$$

In this method, for fixed $\mathbf{Z}_m$, g_k, and β_k, we optimize $\mathbf{W}_m, \mathbf{V}_m$, and $\mathbf{Q}_m$ via alternating approach until convergence. An efficient algorithm is introduced in [8] for a different scenario to solve the inner loop in [8, Algorithm 1]. We employ a similar approach, which for clarity is summarized in Algorithm 1.

Algorithm 1

Input: SIC level, **Output:** $\mathrm{R}_{\mathrm{sum}}$

1: Initialize $\mathbf{W}_m, \mathbf{Q}_m$:

to satisfy the RUs power constraints and to calculate g_k for the first time [8, (31), (32), (33)]

2: Set $i = 0$ **do**

  3: Calculate g_k from (3.3) and $\beta_k = 1/\mathrm{E}_k$,

  4: Calculate $\mathbf{Z}_m = (\mathbf{W}_m \mathbf{W}_m^{\mathrm{H}} + \mathbf{Q}_m)^{-1}$

  5: Update $\mathbf{W}_m, \mathbf{Q}_m, \mathbf{V}_m$ via (3.9),

  6: Calculate $\mathrm{R}_{\mathrm{sum}}^{i}$ via (2.15),

  7: Update $i = i + 1$

while $(R_{\mathrm{sum}}^{i} - R_{\mathrm{sum}}^{(i-1)} > \varepsilon)$;

3.3.1 Computational Complexity

It is worth noting that the optimization problem in (3.9) is a convex semi-definite programming and using the interior-point method it has the worst-case computational complexity of $\mathbf{O}(\max\{n,m\}^4 \sqrt{n} \log(\frac{1}{\varepsilon}))$, where n is the number of variables, m is the number of constraints, and ε is the solution accuracy [44]. Hence, with the assumption of $\varepsilon = 0.01$ and complex-valued multiplication, the computational complexity of Algorithm 1 is at least $\mathbf{O}(4^5 n^{4.5})$.

Notwithstanding the advantages of the proposed optimization for CRAN, computational complexity is still a significant challenge in realizing practical CRAN networks. Hence, we incorporated machine learning techniques to overcome these challenges. Chapter 4 is dedicated to the application of deep neural networks in CRAN optimization.

3.4 Numerical Results

In this section, we provide numerical results from system simulations to examine our algorithm and compare its performance against the state-of-the-art time-division hybrid RF/FSO method proposed in [8]. In the simulations, the same setup as in [8] is adopted. The CRAN scenario uses $M = 2$ RUs each equipped with N^{RU} RF transmit and receive antennas and one optical receiver. The CP node is equipped with $N^{\mathrm{CP}} = 10$ RF transmit antennas and two FSO transmitters, each dedicated to one RU. We consider $K = 4$ users, and all have the same contribution

Table 3.1: FSO channel weather parameters [1, Table I].

Atmospheric loss			GG parameters	
ID	Weather	σ_d (dB/km)	Turbulence	(a,b)
L1	clear	0.43	Strong	(8.05, 1.03)
L2	Haze	4.2	Moderate	(2.23, 1.54)
L3	Fog	20	Weak	(17.13, 16.04)

in our weighted sum-rate objective function (2.15). According to [26], we consider SIC levels of between 45 dB to 113 dB for IBFD at the RUs. Three weather conditions are investigated for the FSO channel, referred to as L1, L2, and L3, and shown in Table 3.1. In the following, we show results as a function of the SIC level to account for different IBFD solutions.

First, Figure 3.1 shows the sum-rate of the proposed approach as a function of SIC under different weather conditions and various RF bandwidths. The most stable performance occurs in the L1 weather condition where FSO is fully functional. We further observe that for this weather condition and RF bandwidths of 20 and 40 MHz, the sum-rate does not change much with the SIC level. This is because the FSO link provides sufficient backhaul capacity and highly accurately SIC is not required. In case of the 80 MHz bandwidth, we note that the extra backhaul capacity provided by the RF link improves the user's sum-rate even in the L1 weather condition. This emphasizes the importance of hybrid RF/FSO for very high data-rate communication. Next, in Figure 3.2 we highlight the benefits of the FD transmission compared to HD considered in [8] in terms of sum-rate. For the HD case, we show the results achieved under the best time allocation between RF backhaul and user links. We observe that the proposed FD-based method outperforms the HD transmission given a sufficient SIC level. The intersection point between the respective curves is at fairly benign SIC levels of 60-70 dB, and independent of the RF signal bandwidth. Better weather conditions (i.e., L1 vs L2 and L2 vs L3) render the FD solution beneficial at lower SIC levels, as the FSO link can provide more of the backhaul capacity in the hybrid RF/FSO system.

Finally, Figure 3.3 demonstrates the interplay between signal attenuation in the backhaul link due to CP-to-RU distance and SIC levels. Clearly, highly effective

Table 3.2: Simulation Parameters.

Parameter	Symbol	Value
Number of antennas at the CP	N^{CP}	10
Number of RUs	M	2
Number of antennas at RU	N^{RU}	4
Number of users	K	4
Distance from the CP to the RUs	d_{fh}	1 km
Distance from the RUs to the users	d_{dl}	100 m
Parameters of the FSO links		
Parameter	Symbol	Value
Transmit power of FSO transmitter	P_{FSO}	10 dBm
FSO signaling rate	B^{FSO}	1 Gbaud
Divergence angle of the laser beam	ϕ	2 mrad
Radius of the receiver aperture	r	10 cm
Responsivity of the photodetector	R	0.5 A/W
Noise variance at the receiver	σ^2_{FSO}	$10^{-13}\mathrm{A}^2$
Parameters specific to the RF backhaul link		
Parameter	Symbol	Value
Transmit power of the CP	P_{cp}	33 dBm
Breakpoint distance for the FH link	$d^{\mathrm{fh}}_{\mathrm{break}}$	100 m
Line-of-sight pathloss exponent	n_{LoS}	2.5
Rice factor (Rician fading factor)	K_r	5 dB
Antenna gains for the FH link	$(G_{\mathrm{CP}}, G_{\mathrm{RU}})$	(3dBi, 3dBi)
Parameters specific to the RF downlink		
Parameter	Symbol	Value
Transmit power of the mth RU	P_m	23 dBm
Breakpoint distance for the downlink	$d^{\mathrm{dl}}_{\mathrm{break}}$	10 m
Non-line-of-sight pathloss exponent	n_{nLoS}	3.5
Antenna gains for the downlink	$(G_{\mathrm{RU}}, G_{\mathrm{MU}})$	(3dBi, 3dBi)
RF parameters		
Parameter	Symbol	Value
Carrier frequency	f_c	3.6 GHz
Bandwidth of the RF signal	B^{RF}	20,40,80 MHz
Noise power spectral density	N_0	-170 dBm/Hz
Noise figure of the receivers	N_{F}	7 dB

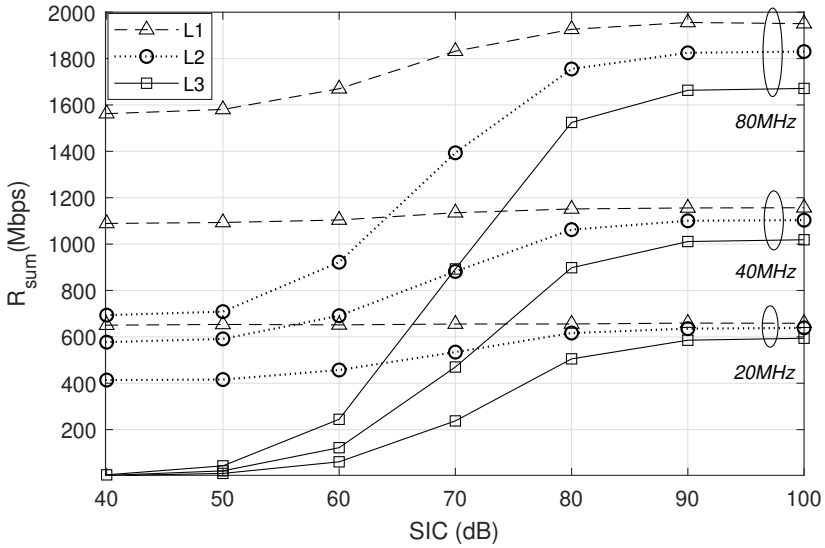

Figure 3.1: Average sum-rate versus SIC level for different weather conditions and RF bandwidths.

interference cancellation in IBFD at levels of 90 dB or more provides fairly robust sum-rate performance as the backhaul link distance increases. On the other hand, systems with low SIC levels experience significant degradations due to the effect of the RU transmit signal on its received signal in the more attenuated RF backhaul link.

3.5 Conclusion

In chapters 2 and 3, we studied the problem of IBFD hybrid RF/FSO in a CRAN architecture containing multiple RUs and users. We formulated the problem of designing the CP and RU beamforming and quantization vector at RU as a sum-rate maximization problem. We approximated the derived non-convex optimization problem using a semi-definite convex optimization problem by manipulating objective function and capacity constraint. We solve this problem by alternating convex optimization and provide a lower-bound for the user's sum-rate. The proposed method was simulated under different weather conditions and bandwidths, and it is shown to outperform the state-of-the-art time-division hybrid RF/FSO approach

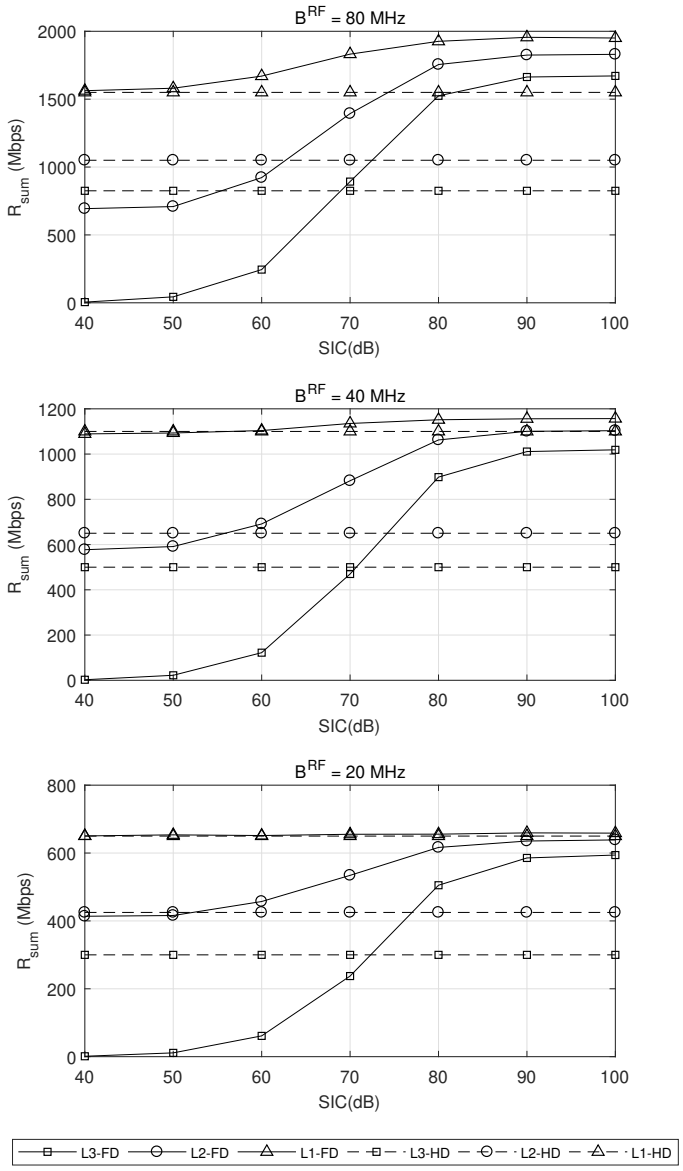

Figure 3.2: Average sum-rate versus SIC level for HD and FD RUs. The RF bandwidth is 20 MHz, 40 MHz and 80 MHz, and the three weather conditions L1, L2 and L3 are considered.

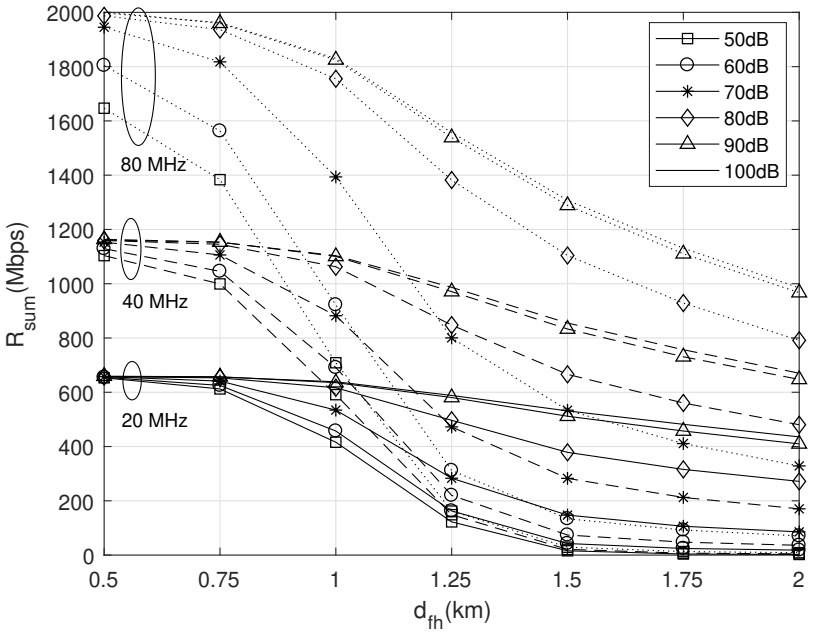

Figure 3.3: Average sum-rate for versus the distance of the backhaul link for different SIC levels. L2 weather condition.

provided sufficient SIC.

Chapter 4

Machine Learning-Based CRAN Optimization, a Supervised Approach

4.1 Learning to Optimize

For years, a combination of analytical modeling and heuristic numerical approaches played a vital role in designing and optimizing wireless communication systems. However, this method is not as effective in solving more complicated problems that also are time-critical. With the rise of 5G and the unprecedented increase in the number of nodes and users in the network and strict low-latency requirements of many 5G applications, heuristic optimization benefits are curbed. In recent years, many scientist across different fields have started integrating the machine learning techniques in complicated optimization problems [45–47]. This trend is also seen in telecommunications, e.g., spectrum sensing [48], relay selection in high-mobility networks [49], anomaly detection for security in RAN [50], resource allocation in 5G [51], MIMO beamforming [52, 53] and channel estimation [54, 55].

In this study, we employed DNNs to solve the optimization problem introduced in the previous section.

4.1.1 Why and When Should We Use The Neural Network?

In case the nature of an optimization problem requires constantly re-solving the optimization for different sets of input parameters, knowing a mapping from the set of parameters to the optimal solution would be extremely helpful. This way, instead of running an optimization algorithm online, to find the optimal solution for each input parameter, we merely need to feed the parameters to the mapping function and obtain the solution with low computational complexity. Consider the following unconstrained optimization problem

$$\min_{x} \; f(x;p), \tag{4.1}$$

where p is the set of parameters of the problem. If p varies quite often, instead of solving this problem numerically each time, we would rather find the mapping between the optimal point (x^*) and p. In such a case, a neural network can be trained in an unsupervised or supervised manner to find the mentioned mapping.

In the CRAN optimization problem (3.9) in chapter 2, we jointly optimized three variables—access-link linear precoders ($\mathbf{W} = [\mathbf{W}_1, ..., \mathbf{W}_{N_{\mathrm{RU}}}]$), backhaul linear precoder ($\mathbf{V} = [\mathbf{V}_1, ..., \mathbf{V}_{N_{\mathrm{RU}}}]$), and CP quantizer ($\mathbf{Q} = [\mathbf{Q}_1, ..., \mathbf{Q}_{N_{\mathrm{RU}}}]$), in order to maximize the weighted sum-rate of the users. The optimal solution of our problem is a function of backhaul and access link's Channel State Information (CSI). Since CSI is a fast-changing parameter in wireless communication, the aforementioned optimization task requires to be constantly re-solved upon any change in CSI. Hence, in this problem finding the mapping from the CSI to the optimal $\mathbf{W}$, $\mathbf{V}$, and $\mathbf{Q}$ via a DNN is very helpful. Figure 4.1 shows the train and test phases of this method.

In case we can successfully train this neural network, we can overcome the computational complexity of the semi-definite programming problem (3.9) and drastically reduce the required time to find the optimal solution. One major issue, however, in our case, is the presence of multiple constraints in the optimization problem. We must enforce the output of the DNN to produce solutions that lie in the intersection of the RU power constraints, CP zero-forcing constraint, backhaul capacity limitation, and positive semi-definiteness of $\mathbf{V}$ and $\mathbf{Q}$ matrices and at the same time accomplish an acceptable performance in terms of users sum rate.

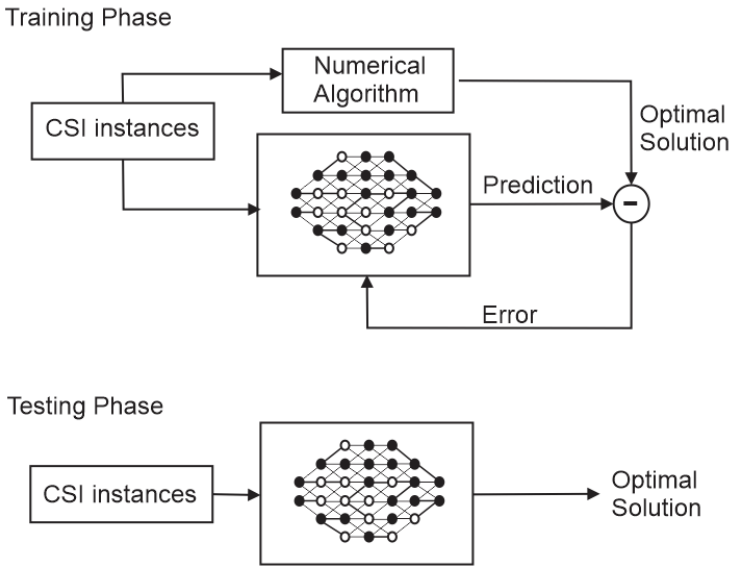

Figure 4.1: Learning to optimize. Supervised learning train and test phases.

4.1.2 How An Artificial Neural Network Works?

An Artificial Neural Network (ANN) is a computing system that consists of connected nodes forming multiple linear layers, each of which equipped with a non-linear activation function. The structure of an ANN is illustrated in Figure 4.3. It can approximate complicated functions given a large enough data set, enough number of layers and proper selection of parameters and hyper-parameters. In the training phase, the ANN learns the features of input data and finds the linear transformation required to generate the next set of features to feed the next layer. The activation function provides the non-linearity required to increase the model's flexibility. If X is the input, and Y is our label vector, using supervised learning notation form [56], we have:

$$X = \begin{bmatrix} x_1^{\mathrm{T}} \\ x_2^{\mathrm{T}} \\ \vdots \\ x_n^{\mathrm{T}} \end{bmatrix}_{n \times d} , \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}_{n \times 1} , \quad \Delta = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_k \end{bmatrix}_{k \times d} . \tag{4.2}$$

In order to handle the bias-variable or y-intercept, and take into account the non-linearity of model, linear latent-factor features of input, $z_i = \Delta x_i$, should go through a non-linear function called activation function, where

$$Z = \begin{bmatrix} z_1^{\mathrm{T}} \\ z_2^{\mathrm{T}} \\ \vdots \\ z_n^{\mathrm{T}} \end{bmatrix}_{n \times k} \quad , \quad L = \begin{bmatrix} l_1 \\ l_2 \\ \vdots \\ l_k \end{bmatrix}_{k \times 1} \quad . \tag{4.3}$$

Using linear model L, ANN model will make a prediction

$$y_i = L^{\mathrm{T}} \mathrm{h}(z_i) \tag{4.4}$$

that minimizes the loss function. This process happens at all hidden layers. Figure 4.2 illustrates one hidden layer of a neural network. During the training phase, ANN will repeatedly update the L and Δ jointly, until it gets to the output layer. Each neuron can detect a portion of feature and as the data flows in the layers, deeper layers can detect a combination of features. By definition, an ANN with more than two hidden layers is a deep neural network.

4.2 Data Collection and Pre-processing

4.2.1 Data collection

In supervised learning, we need labeled training data to train the network. Generally, such labeled data can be obtained either from mathematically solving the complicated problem many times or from a real mobile network or a combination of two.

In our case, the labeled data is a large set of backhaul and access link CSIs as input features and the corresponding optimal solutions obtained from CRAN optimization as input labels, gathered from solving the heuristics optimization approach introduced in the previous chapter. Considering the large computational complexity of solving the semidefinite programming (3.9) introduced in chapter 2, the time required for gathering a large-enough data-set is very long.

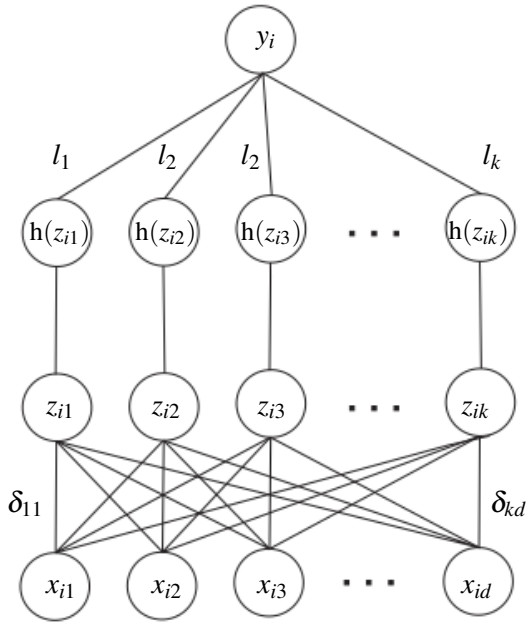

Figure 4.2: Hidden layers of deep neural network.

Table 4.1: Compute Canada resources [2].

Nodes	CPU (GHz)	RAM	storage	number of runs	run-time
56	2 x Intel @ 2.1	125GB variable	2 x 480G SSD	20000	3 weeks

To recap, the computational complexity of the numerical algorithm 1 we used for solving the CRAN optimization (3.9) is $\mathbf{O}(2n^{4.5})$. To provide enough computational resources for solving this algorithm tens of thousands of times, we used the Compute Canada CPU resources. Besides, to make the algorithm faster, we used the parallel processing capability of MATLAB® using *parfor* that "executes for-loop iterations in parallel on workers in a parallel pool" [57], which reduced the amount of processing time considerably. For details on computational resources and run-time refer to Table 4.1.

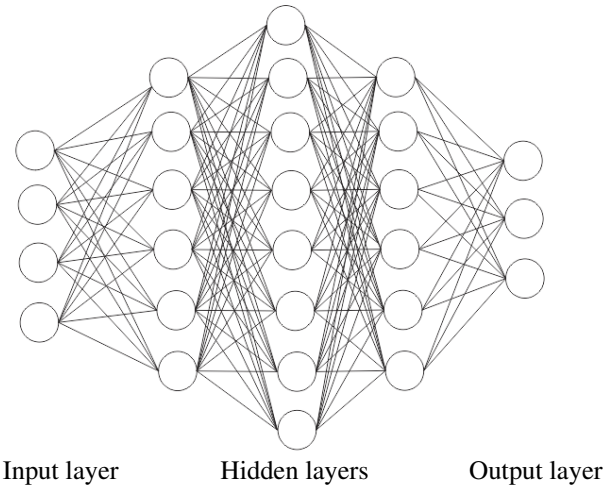

Figure 4.3: A DNN with three hidden layers.

4.2.2 Dealing with Complex-Valued Tensors

One of the primary challenges of using deep learning in our problem is dealing with complex numbers. In our problem five complex-valued tensors—backhaul CSI, access-link CSI, backhaul precoding matrix, access-link precoding matrix, and CP quantization noise covariance matrix are involved. In Table 4.2, the details of the size of these tensors are provided.

In neural networks, one way to deal with complex-valued tensors is to separate the real and imaginary parts and stack them or concatenate them to end up with real-valued tensors that DNNs can use for training. Although such a pre-processing potentially can obscure the relation between real and imaginary parts of the data by treating them as unrelated information in general, this relation can be learned in the case of fully connected DNNs. This pre-processing has been broadly used in data-driven telecommunication articles [58–60].

In the case of the Convolutional Neural Network (CNN), stacking or concatenating the real and imaginary parts of the tensors can cause information loss as it breaks some existing geometrical relations and creates new ones. To prevent this, a new type of DNNs, called deep complex neural network, is proposed by

Table 4.2: Data-set dimensionality.

Parameter	Raw Complex Data	Flatten Real Data
$\mathbf{W}$	$20000 \times N_{tx}^{RU} \times N_{tx}^{RU} \times M$	$2 \times 20000 \times (N_{tx}^{RU} N_{tx}^{RU} M)$
$\mathbf{Q}$	$20000 \times N_{tx}^{RU} \times N_{tx}^{RU} \times M$	$2 \times 20000 \times (\frac{N_{tx}^{RU} + N_{tx}^{RU}(N_{tx}^{RU}-1)}{2} M)$
$\mathbf{V}$	$20000 \times N_{tx}^{CP} \times N_{tx}^{CP} \times M$	$2 \times 20000 (\frac{N_{tx}^{CP} + N_{tx}^{CP}(N_{tx}^{CP}-1)}{2} M)$
$\mathbf{CSI_{DL}}$	$20000 \times N_{tx}^{RU} \times K \times M$	$2 \times 20000 \times (N_{tx}^{RU} N_{tx}^{RU} M)$
$\mathbf{CSI_{FL}}$	$20000 \times N_{tx}^{CP} \times N_{tx}^{RU} \times M$	$2 \times 20000 \times (N_{tx}^{CP} N_{tx}^{RU} M)$

[61]. However, in this study we only use fully-connected DNNs, thus we use the conventional real-imaginary concatenation method.

4.2.3 Dealing with High-Dimensional Tensors

To feed the neural network with the data gathered from our problem, we need to flatten both the input features and input labels, as shown in Figure 4.4. Flattening the data for training fully connected DNNs is easy, as the reshaping order does not matter. However, to be able to use the output of the neural network, we need a function that recovers $\mathbf{V}$, $\mathbf{W}$, and $\mathbf{Q}$ from their flat real vector. Therefore, it is important to use a transformation that can undo itself in the reverse direction.

There are three standard reshaping orders for flattening tensors—C, A, and F[1]. Among these orders, only the order F has this property. Meaning that, for example, if we reshape a 3D tensor $\mathbf{W}$ of size $a \times b \times c$ to a vector of size $1 \times abc$ with order F and then reshape it again to the size $a \times b \times c$, we get the same $\mathbf{W}$. However, it is not true for order C and A.

4.3 Neural Network Model

DNNs and CNNs are among the most frequently used models in multi-antenna system designs. Although DNN is a more general approach that targets a broader

[1] According to [62]: "C means to read/write the elements using C-like index order, with the last axis index changing fastest, back to the first axis index changing slowest. 'F' means to read/write the elements using Fortran-like index order, with the first index changing fastest, and the last index changing slowest. 'A' means to read / write the elements in Fortran-like index order if the array is Fortran contiguous in memory, C-like order otherwise."

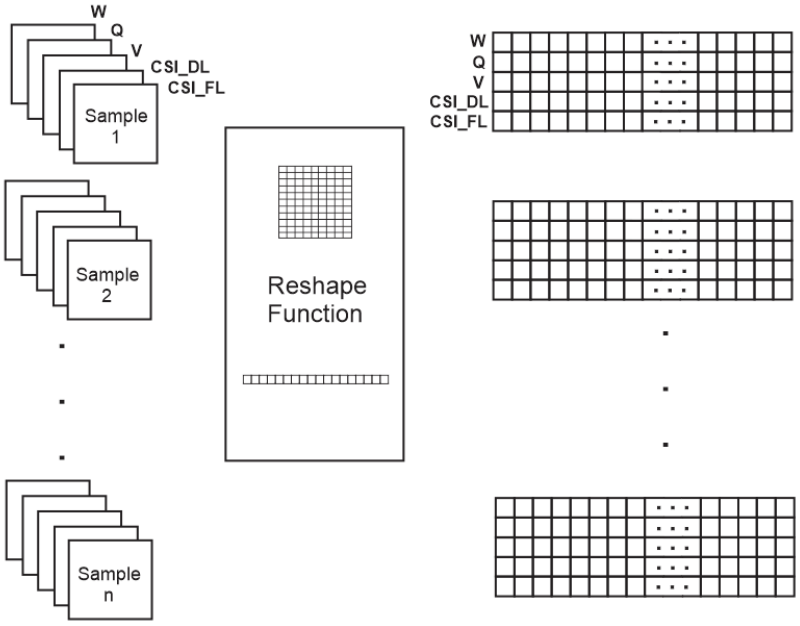

Figure 4.4: Flattening the data.

range of applications, CNN is specialized in learning geometrical patterns and relations.

There are no geometrical relations or local inter-dependencies in backhaul and access link CSI matrices in our problem. That is because there is no actual semantic relation between adjacent rows or columns, and just by re-ordering the indices of the antenna ports, we can obtain a new equivalent representation of the CSI matrix. In other words, changing the order of rows or columns of a CSI matrix does not add or remove any information.

Hence, a fully connected DNN is a more appropriate option for our problem compared to a CNN. More elaborate details about the structure of the DNN are discussed in Section 4.4 ,and the post-processing steps required for enforcing constraints will be provided in Sections 4.5 and 4.5.2.

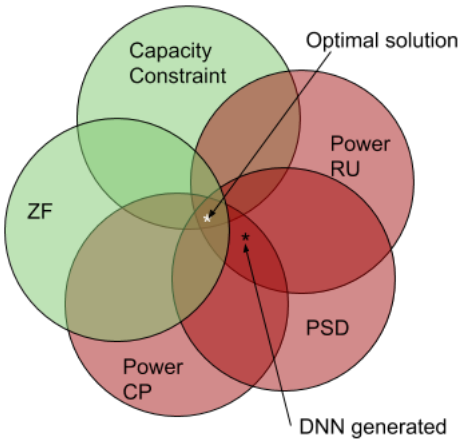

Figure 4.5: The Venn diagram of the constraints. Note that Failing to meet the capacity constraint and ZF constraint does not make a DNN-generated $[\mathbf{V}, \mathbf{W}, \mathbf{Q}]$ solution infeasible, though affects the sum rate. However, failing to meet the Positive Semi-Definite (PSD) and power constraints leads to an infeasible solution.

4.4 Performance Metrics

Achieving an appropriate DNN architecture for our problem requires several parameters and hyper-parameters tuning steps. Some of the parameters and hyper-parameters to be designed in a DNN are the number of hidden layers, the activation functions, batch size, initialization types, number of epochs, number of nodes in each layer, optimization configuration such as the optimizer, learning rate, drop out rate, regularization type, and value, etc. Deciding on parameters and hyper-parameters types and values is a decision based on the the DNN's performance. In most DNNs, the train and validation errors and their slopes are useful in determining both the parameters and hyper-parameters. Though, in our case, the loss value is not the only factor to be considered. The loss function measures how close we are to the input labels in the data-set. However, it does not guarantee the feasibility of the solution and can not give us a good sense of the sum-rate performance in our case. As we get closer to the input labels, more predictions satisfy the constraints and better performance we get, though we should observe all performance metrics to decide on a proper DNN architecture.

During the training, our implicit goal is to maximize the user sum-rate given in (2.15), so one performance metric is the amount of sum-rate achieved by the neural network. Also, since we cannot guarantee that the DNN-generated solutions can satisfy the constraints, another performance metric is the ratio of feasible solutions to all DNN-generated solutions and the amount of violation from the feasible set. Only all of these metrics together can measure our network's performance. With proper observation and evaluation of all performance metrics together with the loss values, we can land on a good DNN architecture that leads to a low validation error, good performance, and feasible predictions.

As discussed before, we cannot make sure all the predictions are feasible. In the next section, we will discuss the post-processing steps needed to enforce constraints in our predictions.

4.5 Accounting for Constraints

Our goal is to train a neural network that takes the backhaul and access link CSI matrices and produces $\mathbf{V}$, $\mathbf{W}$, and $\mathbf{Q}$ such that, first, satisfies the power, ZF, capacity, and PSD constraints and second, maximize the user sum-rate formulated in (3.9).

In this chapter, we approach this problem with a supervised learning method. In other words, we will train a neural network that behaves, just like the optimization method we introduced in chapter 3. This way, by learning to produce a solution set $[\mathbf{V}, \mathbf{W}, \mathbf{Q}]$ per CSI that is similar to the solution set achieved by the optimization algorithm for the same CSI, it indirectly learns to maximize the sum rate.

However, the presence of multiple constraints in this problem causes many challenges. By mimicking the characteristics of optimal solutions in the data-set, we cannot ensure that all constraints will be met. Figure 4.5 illustrates this by a Venn diagram. Failing to meet the ZF constraint causes interference on RUs, which affects the backhaul capacity and, consequently, the sum-rate, but it does not make a DNN-generated solution infeasible. The same is true for the backhaul capacity constraint; if the sum-rate surpasses the backhaul capacity, it causes congestion, and the final sum-rate becomes capped by the backhaul capacity, but it does not make a solution infeasible. However, failing to meet the power constraints on $\mathbf{V}$

and $\mathbf{W}$ or the PSD constrain on $\mathbf{V}$ and $\mathbf{Q}$ makes a $[\mathbf{V}, \mathbf{W}, \mathbf{Q}]$ solution infeasible. To overcome these challenges, we adopt various techniques for each constraint and discus them in the next sections.

4.5.1 Backhaul ZF and Capacity Constraints

In supervised learning, the DNN implicitly learns how to avoid large interference as all of the input labels that are provided for training satisfy the ZF constraint; however, since we cannot guarantee that the DNN can meet the ZF constraint, we consider the adverse effects of the residual interference in terms of user sum rate.

Failing to satisfy the ZF constraint causes interference $\mathbf{G}_m^{\mathrm{H}}\mathbf{V}_k\mathbf{G}_m$ that affects the RF backhaul capacity as follows:

$$C_m^{\mathrm{RF}}(\mathbf{V}_m, \mathbf{Q}_m, \mathbf{W}_m) = \ln\det\left(\mathrm{I} + (\mathbf{G}_m^{\mathrm{H}}\mathbf{V}_k\mathbf{G}_m + \alpha\mathbf{P}_m^{\mathrm{si}} + \sigma_n^2\mathbf{I})^{-1}\mathbf{G}_m^{\mathrm{H}}\mathbf{V}_m\mathbf{G}_m\right),\ k \neq m \tag{4.5}$$

The larger the $\mathbf{G}_m^{\mathrm{H}}\mathbf{V}_k\mathbf{G}_m$ becomes for $k \neq m$, the more interference it causes and further limits the RF backhaul capacity and consequently mitigates the user sum rate by causing congestion.

However, as indicated in Figure 4.5, violating the PSD constraint of $\mathbf{V}$ and $\mathbf{Q}$ and power constraint of $\mathbf{V}$ and $\mathbf{W}$ makes a solution infeasible and useless. Hence, we enforce the PSD and power constraints through the structure of the DNN.

4.5.2 Enforcing PSD Constraints

A positive semi-definite matrix should be Hermitian. To enforce the Hermitian property, we employ two separate DNN's to generate the diagonal and upper triangular elements of the covariance matrices $\mathbf{Q}_m$ and $\mathbf{V}_m$ as seen in Figure 4.6. For example, for the matrix

$$\mathbf{Q}_m = \begin{bmatrix} \mathrm{q}_{11} & \mathrm{q}_{12} & \mathrm{q}_{13} & \mathrm{q}_{14} \\ \mathrm{q}_{21} & \mathrm{q}_{22} & \mathrm{q}_{23} & \mathrm{q}_{24} \\ \mathrm{q}_{31} & \mathrm{q}_{32} & \mathrm{q}_{33} & \mathrm{q}_{34} \\ \mathrm{q}_{41} & \mathrm{q}_{42} & \mathrm{q}_{43} & \mathrm{q}_{44} \end{bmatrix}, \tag{4.6}$$

we first decompose it into two arrays of diagonal $\mathbf{Q}_m^{\text{diag}} = [q_{11},\ q_{22},\ q_{33},\ q_{44}]$ and upper-triangular $\mathbf{Q}_m^{\text{upper}} = [q_{12},\ q_{13},\ q_{14},\ q_{23},\ q_{24},\ q_{34}]$ and then train a separate neural network for each vector. This way, the Hermitian property of the generated $\mathbf{V}_m$ and $\mathbf{Q}_m$ are enforced. The structure of this neural network is illustrated in Figure 4.6.

To ensure the positive semi-definiteness of the DNN-generated $\mathbf{V}_m$ and $\mathbf{Q}_m$ matrices, after training, we pass the generated $\mathbf{V}_m$ or $\mathbf{Q}_m$ matrices to a customized function that approximates the generated tensors by their closest semi-definite matrices. To do so, after the training, we first combine $\mathbf{Q}_m^{\text{diag}}$ and $\mathbf{Q}_m^{\text{upper}}$ to get $\mathbf{Q}_m$. In the mentioned activation function, we perform eigen-value decomposition on $\mathbf{Q}_m$ as

$$\mathbf{Q}_m = \Delta\Sigma\Delta^{\text{H}}, \tag{4.7}$$

where, Σ is a diagonal matrix containing the eigen-values of $\mathbf{Q}_m$. Then we approximate Σ with a Σ' with all negative elements replaced by 0 and re-construct the PSD-approximated of $\mathbf{Q}_m$ as follows:

$$\mathbf{Q'}_m = \Delta\Sigma'\Delta^{\text{H}}. \tag{4.8}$$

This way we enforce the PSD constraint. The same structure can be applied to $\mathbf{V}_m$ matrix to enforce the PSD constraint.

4.5.3 Enforcing Power Constraint

To enforce the $\mathbf{W}_m$ and $\mathbf{Q}_m$ to meet the power constraints after training the neural network we employ a customized function in the output layer of the neural network to first normalize the vector and then scale its power to the maximum allowed power as suggested by [60]. So the customized activation function of the output layers are

$$\mathbf{W}_m^{scaled} = \mathbf{W}_m \sqrt{\frac{\text{P}_m - |\text{Tr}(\mathbf{Q}_m)|}{|\text{Tr}(\mathbf{W}_m\mathbf{W}_m^{\text{H}})|}}, \tag{4.9}$$

and

$$\mathbf{Q}_m^{scaled} = \mathbf{Q}_m \frac{\text{P}_m - |\text{Tr}(\mathbf{W}_m\mathbf{W}_m^{\text{H}})|}{|\text{Tr}(\mathbf{Q}_m)|}. \tag{4.10}$$

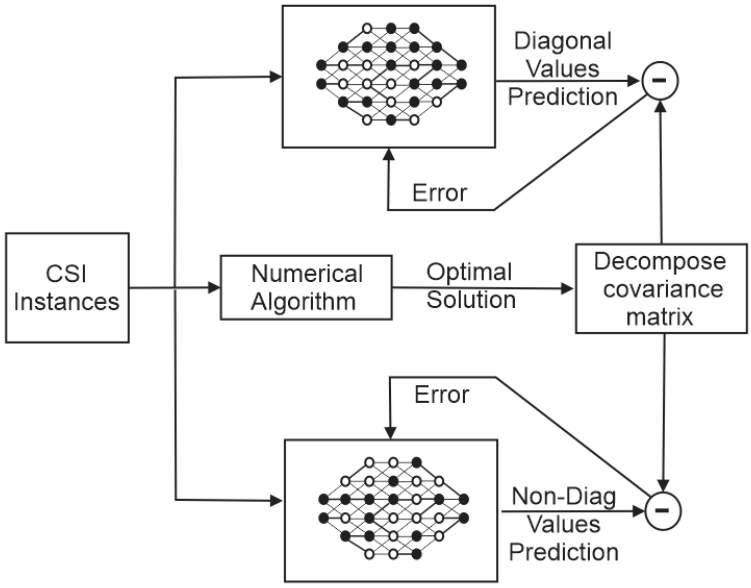

Figure 4.6: The proposed DNN structure to ensure Hermitian output.

For backhaul power constraint, we use a similar technique but since the power constraint is on the $\sum_m \mathrm{Tr}(\mathbf{V}_m)$, we need to keep the ratio of transmit power between different m before and after the scaling. For $m = 2$, the pre-scaling power ratio is

$$\alpha_v = \frac{\mathrm{Tr}(\mathrm{V}_1)}{\mathrm{Tr}(\mathrm{V}_2)} = \frac{\mathrm{Tr}(\mathrm{V}_{\mathrm{scaled}})}{\mathrm{Tr}(\mathrm{V}_{\mathrm{scaled}})}. \tag{4.11}$$

Assuming $\Sigma_{m=1}^{M} \mathrm{Tr}(\mathbf{V}_m) = \mathrm{P}_c$, a proper power scaling that enforces the backhaul RF power constraint and keeps the power ratio is

$$\mathbf{V'}_1 = \mathbf{V}_1 \frac{\alpha_v \mathrm{P}_c}{(1+\alpha_v)\mathrm{Tr}(\mathbf{V}_1)}, \tag{4.12}$$

$$\mathbf{V'}_2 = \mathbf{V}_2 \frac{\mathrm{P}_c}{(1+\alpha_v)\mathrm{Tr}(\mathbf{V}_2)}. \tag{4.13}$$

With these post-processing actions, we can make sure all DNN predictions are in the feasible set. In the next section, we will summarize the DNN performance and architecture.

4.6 Implementation and Numerical Results

4.6.1 PyTorch

PyTorch is one of the most powerful machine learning libraries used to create artificial neural network models. It is recognized for its descriptive documentation and active developer community. We used four packages in PyTorch to implement our neural network—PyTorch DNN model, automatic differentiation to compute gradient, PyTorch loss to compute loss function and PyTorch optimizer to optimize the model parameters. All of these packages can be customized and incorporated with user-defined modules if there is a specific demand in the problem that cannot be met by these packages (as we do so in our unsupervised learning model in Chapter 5).

During the training phase, the neural network tries to minimize loss through parameter optimization and gradient calculation is an essential part of the process. In PyTorch the data-type of the inputs and outputs of the neural network as well as the wights and biases are tensor. To calculate the gradient of the loss function with respect to the weights and biases, we can activate the *auto-grad* attribute of the weight and bias tensors. This way, when we do an operation in the forward path, PyTorch forms a computational graph which automatically calculates the partial gradients and using the chain rule, obtains the gradient vector to be used in the backward path.

4.6.2 Results

In this subsection, we present the performance results of the proposed supervised learning approach for solving the CRAN optimization problem. Each DNN can predict one optimization variables in a specific weather condition and SIC level. For each weather condition and SIC level, we gathered a data-set containing 20000 labeled samples, including 16000 training samples and 4000 validation samples. The main difference between each scenario is the availability of the FSO link based on weather conditions and backhaul RF link based on the SIC level. The hyper-parameters of the DNN are listed in Table 4.3 and input and output sizes in Table 4.4.

We divided the training data set into 80 batches of size 200 and used batch gradient and RMSprop[2] for training. We used Mean Average Error (MAE) and Mean Square Error (MSE) loss functions as illustrated in Table 4.3 for different networks. A combination of drop out and L2 regularization techniques is used to solve the over-fitting problem in training phases. Furthermore, we observed a gradient explosion at the final stages of training and solved it via gradient clipping. To evaluate the final sum-rate performance, we used separate DNNs to train **V**, **Q** and **W**. Each of these DNNs can be used to predict one of the optimization variables. The model evaluation can be done in two ways. One approach is to combine each DNN-generated variable with the optimal results obtained in Chapter 3 and calculate the sum rate (let's name it Semi-DNN sum-rate) as illustrated in Figure 4.8. We can compute this metric for each DNN separately. To evaluate the overall performance of all DNNs together, we can compute the sum-rate performance by passing the output of all three DNNs to evaluate the sum-rate (let's name it DNN sum-rate) as shown in Figure 4.7. The results of both approaches are summarized in Table 4.5 and associated DNN design parameters and input/output size are detailed in Tables 4.3 and 4.4. As it is seen from Table 4.5, the overall performance is limited by the performance of **W** prediction. We believe that the mandatory post-processing that we do on the DNN-generated **W**, **Q** and **V** after training to satisfy the PSD and power constraints affects the performance. In the case of **V**, any small divergence of the optimal covariance matrix leaves us with an inter-RU interference in the backhaul, which decreases the backhaul's capacity and caps the final sum-rate. In the case of **Q** and **W**, training the DNN was quite challenging and our best guess is that, compared to **V**, the variables **Q** and **W** have a more complicated behavior and it is more difficult to predict them.

Computational Complexity Comparison

Considering the variables' size in our system model, the computational complexity of model-based approach introduced in chapter 3 is $\mathscr{O}((4^5(2\times(10+4+4))^{4.5}) \approx$

[2] According to [63], In RMSprop "the effective learning rate is divided by square root of gradient average ($\mathrm{lr}_{\mathrm{rmsprop}} = \frac{\mathrm{lr}}{\sqrt{\mathrm{v}+\varepsilon}}$), where ε is an added term to improve the numerical stability of algorithm and $\mathrm{v} = 0.9 \times \mathrm{MeanSquare}(w,t-1) + 0.1(\sigma E/\sigma w(t))^2$, which keeps a moving average of the squared gradient for each weight."

Table 4.3: DNN architecture in supervised learning.

Variable	Loss	Layers & nodes
$\mathbf{W}$	MSE	64, 100, 100, 200, 200, 200, 100, 100, 64
$\mathbf{V_{diag}}$	MAE	160, 200, 200, 100, 50, 20
$\mathbf{V_{tri}}$	MSE	160, 200, 300, 400, 400, 400, 400, 300, 200, 180
$\mathbf{Q_{diag}}$	MAE	64, 100, 50, 25, 15, 8
$\mathbf{Q_{tri}}$	MSE	64, 100, 80, 40, 30, 24

Table 4.4: Inputs and outputs of DNNs in supervised learning approach.

Parameter	Input & Dimension	Output & Dimension
$\mathbf{W}$	$\mathbf{CSI_{DL}}$: 20000×64	$\mathbf{W}$: 20000×64
$\mathbf{V_{diag}}$	$\mathbf{CSI_{FL}}$: 20000×160	$\mathbf{V_{diag}}$: 20000×20
$\mathbf{V_{tri}}$	$\mathbf{CSI_{FL}}$: 20000×160	$\mathbf{V_{tri}}$: 20000×180
$\mathbf{Q_{diag}}$	$\mathbf{CSI_{DL}}$: 20000×64	$\mathbf{Q_{diag}}$: 20000×8
$\mathbf{Q_{tri}}$	$\mathbf{CSI_{DL}}$: 20000×64	$\mathbf{Q_{tri}}$: 20000×24

$\mathscr{O}(10^{10})$. However, the computational complexity of forward path in a DNN is $\mathscr{O}(nm_1 + \sum_{i=1}^{l-1} m_i m_{i+1} + m_l k)$, where n is the number of inputs, l is the number of hidden layers each one with $m_1, m_2, ..., m_l$ neurons and k is the number of outputs. Considering the five DNNs introduced in Tables 4.3 and 4.4, and the average 50% dropout, the overall computational complexity of our supervised data-driven methods is $\mathscr{O}(152800/2) + \mathscr{O}(98000/2) + \mathscr{O}(908000/2) + \mathscr{O}(13145/2) + \mathscr{O}(19520/2) \approx \mathscr{O}(5 \times 10^5)$.

This huge complexity difference between model-based and data-driven optimization, becomes more severe in the next generation of Mobile network where there will be denser network and more networked devices.

Table 4.5: Performance of semi-DNN and DNN architectures.

Variable	Semi-DNN sum-rate	DNN sum-rate
$\mathbf{W}$	75%	
$\mathbf{V}$	82%	73%
$\mathbf{Q}$	99%	

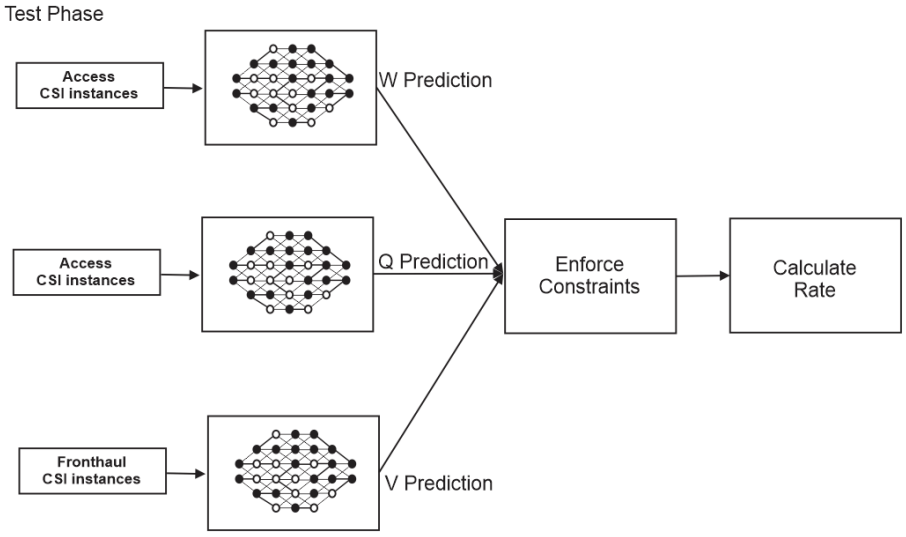

Figure 4.7: DNN-based optimization approach.

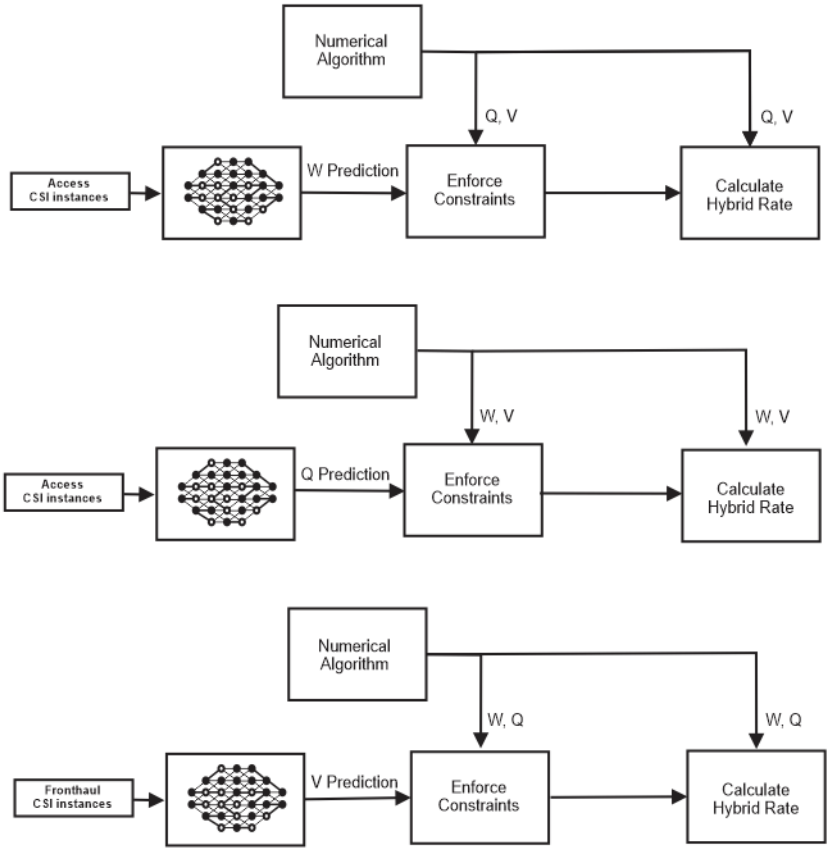

Figure 4.8: Post-processing of DNN-based solutions and achieved hybrid sum-rate.

Chapter 5

Machine Learning-Based CRAN Optimization, an Unsupervised Approach

Notwithstanding the low computational complexity of the supervised DNN-based CRAN optimization introduced in Chapter 4, the performance is limited to the performance of the numerical optimization obtained in Chapter 2. In fact, even reaching to the performance achieved via numerical optimization in Chapter 2 is very difficult and requires arbitrarily large number of epochs and tedious hyper-parameter tuning in the supervised manner.

In addition, the presence of constraints and structural regularization proposed to enforce the constraints (see Figure 3.8), contributed to the inferior sum rate performance achieved in Chapter 4. That is because, the ways we enforced these constraints are not necessarily the most optimal solutions. For example, enforcing the PSD constraint by changing the eigen-values of the DNN-generated $\mathbf{V}$ and $\mathbf{Q}$ can potentially through the solution arbitrarily far away from the optimal solution.

All in all, the sum rate performance achieved in Chapter 4 through the supervised learning is acceptable especially considering the large reduction in computational complexity. However, it seems that employing an unsupervised learning method that is not bounded by the existing solutions in Chapter 2 can potentially achieve a better performance while maintaining a low computational complexity.

Hence in this chapter we study the application of unsupervised learning for CRAN optimization.

5.1 Learning to Optimize with Unsupervised Learning

To enhance the sum-rate performance of the DNN-based solution we can use unsupervised learning to let the DNN explore beyond the existing solutions (input labels) that is considered the ground truth and discover superior solutions in terms of sum-rate while maintaining a low computational complexity. In summary, the benefits of the unsupervised learning in CRAN optimization are

- No need for solving the optimization task (3.9) thousands of times to generate the input labels
- Can potentially surpass the performance achieved in Chapter 4

In this approach, instead of minimizing the conventional MSE/MAE loss function, we set the negative of the objective function equal to the loss function and train a DNN to minimize it [64–66]. Assume that $x^* = \text{argmin}_x f_0(x, p)$ is the optimization problem we want to solve. Similar to the supervised method, we design a DNN as follows:

$$x^* = \text{DNN}(p), \tag{5.1}$$

and the loss is

$$\mathscr{L} = f_0(x, p). \tag{5.2}$$

As an alternative, some employed reinforced learning (RL) to address similar problems. A 2016 study by Kie Li, et.al, uses RL to solve unconstrained continuous optimization problems [45]. In addition, [46, 47] use RL to solve unconstrained discrete combinatorial optimization problems.

However, what has been missing in all of the mentioned works is the constraints. To the best of our knowledge, a method for learning to optimize an objective function under some generic constraints does not exist that ensures meeting the constraints except for the power limitation constraint in beamforming that is usually addressed by a customized activation function that caps the norm of the

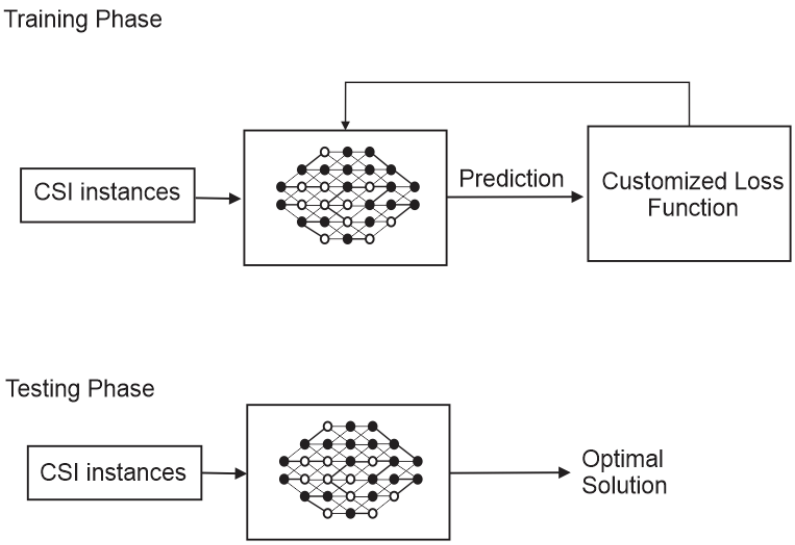

Figure 5.1: Learning to Optimize. Unsupervised learning train and test phases

DNN's output [60]. In this chapter, we introduce a generic method for enforcing any inequality or equality constraints to a DNN-based optimization method.

Assume that $\mathscr{P}$ is the set of all input features, i.e., $p \in \mathscr{P}$ and $\mathscr{X}^*$ is the set of all corresponding outputs of the neural network, i.e., $x^* \in \mathscr{X}^*$. We want this DNN to take samples of p as the input and generate the corresponding x such that it minimizes the the objective function $f_0(x,p)$ under the set of the constraints $f_i(x,p) \leq c_i$ and $h_i(x,p) = b_i$.

5.1.1 Piece-wise Regularization for Enforcing the Constraints

In this novel method, we propose to set the loss function of the DNN equal to the objective function and add penalty terms to the objective function as follows:

$$\mathscr{L}(x^*,p) = f_0(x^*,p) + \sum_i \mathbb{I}(f_i(x^*,p_k) - c_i) + \sum_j \mathbb{I}(h_j(x^*,p_k) - b_j) \qquad (5.3)$$

where, $\mathbb{I}$ is defined as below

$$\mathbb{I}(x) = \begin{cases} 0 & , x \leq 0 \\ \infty & , x > 0 \end{cases}. \tag{5.4}$$

In each epoch, our target would be to minimize the mean of the loss, i.e., $\overline{\mathscr{L}(\mathscr{X}^*, \mathscr{P})} = \frac{1}{|\mathscr{P}|} \sum_{k \in \mathscr{P}} \mathscr{L}(x_k^*, p_k)$. This way, we ensure that the constraints are met as the indicator function disposes of infeasible solutions by sending the value of the loss to infinity. Nonetheless, since the gradient of $\sum_i \mathbb{I}(f_i(x^*, p_k) - c_i) + \sum_j \mathbb{I}(h_j(x^*, p_k) - b_j)$ is always zero, the neural network cannot learn anything with this loss.

A tweak for this issue would be to use penalty terms with non-zero gradient[1] to penalize the infeasible solutions instead of using the $\mathbb{I}$ function. Our target is to make sure that the loss value outside of the feasible set becomes large enough such that no infeasible solution can minimize the $\mathscr{L}$.

Let's define the "deviation from feasibility" for inequality constraints as follows

$$F_i(x^*, p) = \begin{cases} 0 & , f_i(x^*, p) \ \leq c_i \\ \eta_i (f_i(x^*, p) - c_i)^\gamma & , \text{ else} \end{cases} \tag{5.5}$$

and for equality constraints

$$H_j(x^*, p) = \begin{cases} 0 & , h_j(x^*, p) = b_j \\ \eta_j |h_j(x^*, p) - b_j|^\gamma & , \text{ else} \end{cases}. \tag{5.6}$$

where, $\eta_i \geq 0$ are hyper-parameter that tunes the effect of the regulating terms, and $\gamma \geq 1$. Note that η_j must be large enough to dominate $\mathscr{L}$ outside of the feasible set. Now we define the penalty term as follows

$$\Omega(x^*, p) = \sum_i F_i(x^*, p) + \sum_j H_j(x^*, p), \tag{5.7}$$

In this method, for a solution x^*, if none of the constraints get violated, Ω becomes 0 and there is no penalty, otherwise there is a penalty for each constraint

[1] Gradient of the multiplicative penalty term with respect to the weights and biases of the neural network should be non-zero so that the neural network can learn how to avoid infeasible solutions

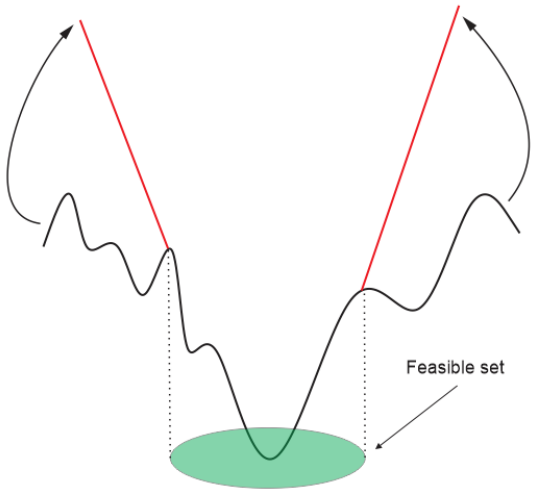

Figure 5.2: Penalizing solutions outside of the feasible set using piece-wise regularization method.

that is violated proportional to the amount of violation (see Figure 5.2). We define the per-sample loss function as follows

$$\mathscr{L}(x^*, p) = f_0(x_k^*, p_k) + \Omega(x_k^*, p_k) \tag{5.8}$$

and the mean loss as follows

$$\overline{\mathscr{L}(\mathscr{X}^*, \mathscr{P})} = \frac{1}{|\mathscr{P}|} \sum_{k \in \mathscr{P}} (f_0(x_k^*, p_k) + \Omega(x_k^*, p_k)) \tag{5.9}$$

5.2 Segmented CRAN Optimization Via Unsupervised Learning

Using the method introduced in Section 5.1 we can formulate our constrained optimization problem and solve through unsupervised learning DNNs. However, training a large DNN that outputs all optimization variables—**V**, **Q**, and **W** can be quite challenging in practice. Larger networks are more prone to over-fit and hyper-parameter tuning can be very difficult and time-consuming.

5.2.1 Segmented Optimization

To overcome the tuning challenge mentioned in Section 5.2, we segment the problem of joint optimization of $[\mathbf{V}, \mathbf{Q}, \mathbf{W}]$, to a segmented optimization tasks. As we have a two-hop communication network consisting of backhaul and access links that connect CP, RU's, and UE's, it also does make sense to separate backhaul and access optimization problems and maximize different objectives, i.e., backhaul capacity and user-sum rate in other optimization problems. We use a DNN for optimizing $\mathbf{V}$, and another DNN for optimizing $\mathbf{Q}$ and $\mathbf{W}$. For training $\mathbf{V}$, the objective is to maximize the backhaul capacity while satisfying the ZF, PSD, and power constraints. For training $\mathbf{W}$ and $\mathbf{Q}$, the objective is to maximize the user sum rate while satisfying the power and PSD constraints.

Besides, in an unsupervised manner, the customized loss function adds a huge complexity cost to training of our neural network as the network will calculate the derivation of the customized loss function in each epoch instead of the MSE or MAE functions which have a less complicated gradient calculation compared to a customized function like equation (5.9). Thus separating networks and having separate loss functions for various optimization variables makes the optimization process more manageable and tuning steps easier.

5.2.2 Accounting for Constraints

For power constraints we will use the piece-wise regularization proposed in section 5.1.1 so that unlike the power scaling method used in Chapter 4, the constraint can get involved in the training process and influence the gradient. The main benefit of the proposed piece-wise regularization is that it allows the neural network to learn how to stay within the allowed boundaries of the problem and not violate the constraints.

In addition to the piece-wise regularization method for imposing the power constraint, we will hard-wire the DNN's structure to satisfy the ZF and PSD constraints without applying any regularizing terms. We will show in Section 5.2.3 and 5.2.4 for the first time, that applying a proper customized activation function we can impose these constraints.

We will not enforce the backhaul limitation constraint during training $\mathbf{Q}$ and

$\mathbf{W}$ as we do not want to limit our results to supervised labels and the derived capacity from supervised manner. We will discuss more about capacity limitation enforcement in post processing step.

5.2.3 Optimization of V in a Semi-Supervised Manner

For training $\mathbf{V}$, because of self-interference, the backhaul capacity also depends on $\mathbf{Q}$, and $\mathbf{W}$. So we use input labels of $[\mathbf{Q}, \mathbf{W}]$, while training $\mathbf{V}$ (see equation (2.6)). That is why we call this training method semi-supervised. We have three constraints in optimizing $\mathbf{V}$ —PSD, ZF, and power constraints. Imposing ZF constraint eliminates multi-RU interference as follows:

$$\mathbf{G}_m^{\mathrm{H}} \mathbf{V}_k \mathbf{G}_m = 0 \ \text{ for } \ k \neq m \tag{5.10}$$

The ZF constraint could not be guaranteed in the supervised CRAN optimization in Chapter 4. Also, to impose the PSD constraint, we resorted to manipulating the $\mathbf{V}_m$ and $\mathbf{Q}_m$ and approximated them by a PSD matrix which caused considerable performance loss. However, in unsupervised learning we can guarantee the satisfaction of both ZF and PSD constraints by designing a customized activation function at the output layer of the DNN.

Assume that the frounthaul beamforming vector to m-th RU is $\mathbf{B}_m$, then we have $\mathbf{V}_m = \mathbf{B}_m \mathbf{B}_m^{\mathrm{H}}$. To ensure that $\mathbf{V}_m$ is always PSD, the DNN can output $\mathbf{B}_m$ in the output layer and a customized activation function can take $\mathbf{B}_m$ and return $\mathbf{V}_m = \mathbf{B}_m \mathbf{B}_m^{\mathrm{H}}$ in the output layer of the DNN. However, this way the generated matrix $\mathbf{V}_m$ does not satisfy the ZF constraint. To satisfy the ZF constraint the matrix $\mathbf{B}_m$ must satisfy

$$\tilde{\mathbf{G}}_m^{\mathbf{H}} \mathbf{B}_m = 0 \quad \forall m \in M, \tag{5.11}$$

where

$$\tilde{\mathbf{G}}_m = [\mathbf{G}_1,\ \mathbf{G}_2,\ \ldots\ \mathbf{G}_{m-1},\ \mathbf{G}_{m+1},\ \ldots\ ,\ \mathbf{G}_M]. \tag{5.12}$$

It means that $\mathbf{B}_m$ must be chosen from the null-space of $\tilde{\mathbf{G}}_{\mathbf{m}}^{\mathrm{H}}$. According to [41], if $\tilde{\mathbf{G}}_m = \mathbf{U}_m \Sigma_m \mathbf{Y}_m^{\mathrm{H}}$ is the Singular Value Decomposition (SVD) of $\tilde{\mathbf{G}}_m$, where $\mathbf{Y}_m = [\mathbf{Y}_m^{(1)},\ \mathbf{Y}_m^{(2)}]$, the columns of $\mathbf{Y}_m^{(2)}$ form an orthogonal basis for the null space of $\tilde{\mathbf{G}}_m$. Hence, we can use $\mathbf{Y}_m^{(2)}$ as a basis for the backhaul beamforming matrix $\mathbf{B}_m$ to

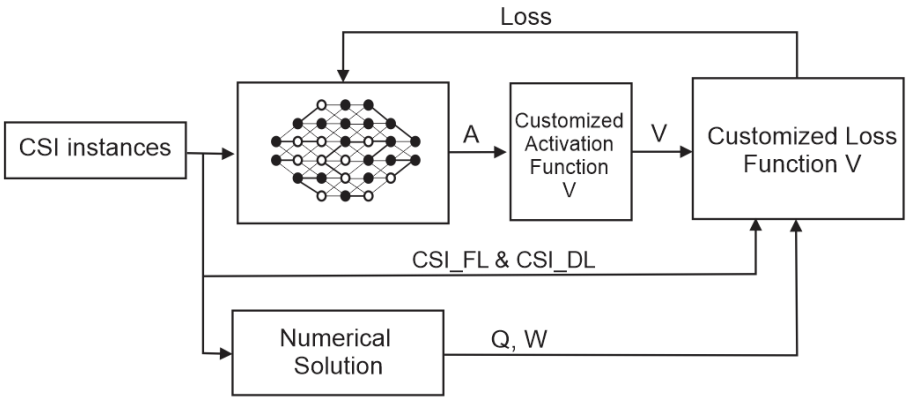

Figure 5.3: DNN structure of the proposed semi-supervised training method for $\mathbf{V}$

satisfy the ZF constraint between the backhaul RF links.

Therefore, using the matrix $\mathbf{Y}_m^{(2)}$ as the basis of the beamforming we define $\mathbf{B}_m = \mathbf{Y}_m^{(2)}\mathbf{A}_m$ where $\mathbf{A}_m$ is a variable matrix that can be optimized for maximizing the backhaul capacity. Now consider a DNN that receives the channel matrices $\mathbf{G} \triangleq [\mathbf{G}_1, \mathbf{G}_2, ..., \mathbf{G}_M]$ and returns $\mathbf{A} \triangleq [\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_M]$ matrices.

$$\mathbf{A} = \mathrm{DNN}(\mathbf{G}) \tag{5.13}$$

For simpler representation, we ignored the tensor flattening and tensor recovering modules here. For this DNN we define the activation function of the output layer of the DNN as follows:

$$\mathbf{V}_m = \left(\mathbf{Y}_m^{(2)}\mathbf{A}_m\right)\left(\mathbf{Y}_m^{(2)}\mathbf{A}_m\right)^{\mathrm{H}} \tag{5.14}$$

This activation function receives $\mathbf{A}_m$ and returns $\mathbf{V}_m$ as according to (5.14). Finally, the customized loss function of the proposed DNN for optimizing $\mathbf{V}$ can be designed based on the piece-wise regularization method proposed in Section 5.1.1. Since in the backhaul our target is to maximize the capacity and we have a power

constraint according to (5.8) we have the per-sample loss as

$$\mathscr{L}_{\mathbf{V}} = -\sum_{m}^{N_{\mathrm{RU}}} \left(B^{\mathrm{FSO}} C_m^{\mathrm{FSO}} + B^{\mathrm{RF}} C_m^{\mathrm{RF}} \right) + \Omega_{\mathbf{V}}, \tag{5.15}$$

where C_m^{FSO} is achieved in (2.7) and C_m^{RF} is achieved in (2.6) and

$$\Omega_{\mathbf{V}} = \begin{cases} 0 & , \sum_m \mathrm{Tr}(\mathbf{V}_m) \ \leq \mathbf{P}_{\mathrm{cp}} \\ \eta \left(\sum_m \mathrm{Tr}(\mathbf{V}_m) - \mathbf{P}_{\mathrm{cp}} \right)^{\gamma} & , \text{ else} \end{cases} \tag{5.16}$$

The structure of the proposed DNN is illustrated in Figure 5.3. As it is seen, during the training the optimal values of $\mathbf{Q}$ and $\mathbf{W}$ are fed to the customized loss function but $\mathbf{V}$ is provided by the DNN. Again, in this type of training since some of the input labels are used and some are not, we name it semi-supervised learning. Also, in this problem we use $\gamma = 2$ and tune η during the training.

5.2.4 Optimization of Q and W in an Unsupervised Manner

For maximizing users sum-rate we propose an unsupervised DNN that received the downlink CSI as input and outputs $\mathbf{Q}$ and $\mathbf{W}$. To meet all the constraints we propose a special DNN architecture as discussed below. The matrix $\mathbf{Q}_m$ must satisfy the PSD constraint. Hence, similar to $\mathbf{V}_m$, our DNN returns an auxiliary variable $\mathbf{S}_m$ and then use a customized activation function to calculate $\mathbf{Q}_m$ as follows

$$\mathbf{Q}_m = \mathbf{S}_m \mathbf{S}_m^{\mathrm{H}}. \tag{5.17}$$

This way, the output of the DNN is always PSD. Both $\mathbf{Q}_m$ and $\mathbf{W}_m$ must satisfy the backhaul capacity and power constraint as indicated in (2.10) and (2.13). We will enforce the capacity constraint in post-processing step by considering congestion for values that violate the capacity constraint. Using the piece-wise regularization method introduced in Section 5.1.1 the customized per-sample loss function of this DNN can be defined as follows

$$\mathscr{L} = -\sum_{m}^{N_{\mathrm{RU}}} R_m^{\mathrm{data}} + \Omega, \tag{5.18}$$

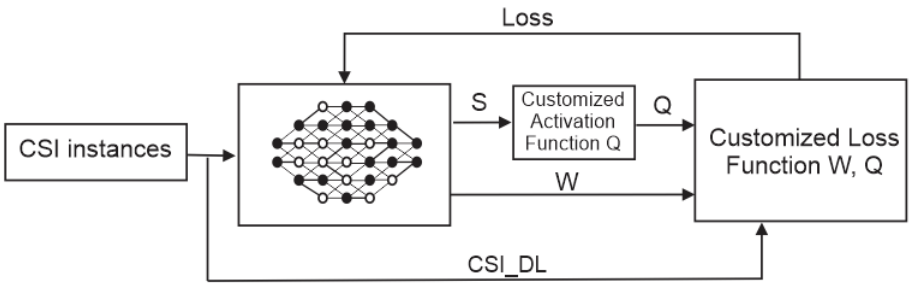

Figure 5.4: DNN structure of the proposed unsupervised training method for **Q** and **W**

where, Ω is the piece-wise penalty term corresponding to the power constraint and is achieved as follows

$$\Omega = \begin{cases} 0 & , \sum_m \mathrm{Tr}(\mathbf{Q}_m) \ \leq \sum_m (\mathbf{P}_m - \mathrm{Tr}(\mathbf{W}_m \mathbf{W}_m^{\mathrm{H}})) \\ \eta(\sum_m \mathrm{Tr}(\mathbf{Q}_m) - \mathbf{P}_m + \mathrm{Tr}(\mathbf{W}_m \mathbf{W}_m^{\mathrm{H}}))^{\gamma} & , \text{ else} \end{cases} . \tag{5.19}$$

The structure of the proposed DNN is illustrated in Figure 5.4. In this problem we use $\gamma = 2$ and tune η during the training.

5.2.5 Post-Processing Step

Penalizing loss function proportional to the amount of violation from constraints, as discussed in Section 5.2.1, can push the DNN to produce values that satisfy constraints, but it cannot guarantee that. So we still need the post-processing step to enforce all predictions to lie in the feasible set. Besides, for constraints that involve multiple variables, it is better to enforce them in the post-processing step. For example, to enforce the capacity constraint, if the sum-rate value derived from unsupervised optimization of **Q** and **W**, violates the backhaul capacity derived from semi-supervised optimization of **V**, we will consider congestion in the post-processing step. This way the minimum of backhaul capacity and sum-rate will be considered as the final sum-rate. Note that we do not need to modify the achieved **W** and **Q** to meet the capacity constraint, as in downlink if the capacity of backhaul is less than the capacity of the access link, users will experience a throughput less

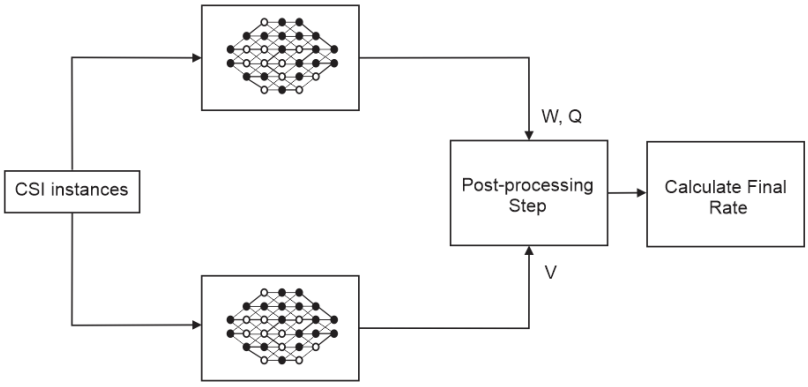

Figure 5.5: Proposed unsupervised and semi-supervised DNNs for CRAN DL optimization.

than their capacity but no data loss will happen.

Enforcing the power constraint in unsupervised manner is slightly different from the supervised method applied in Chapter 4, because now both $\mathbf{W}$ and $\mathbf{Q}$ should be adjusted together to meet the constraint. There are multiple ways to enforce the power constraint—1) by splitting the power budget between outputs of the DNN ($\mathbf{W}$ and $\mathbf{Q}$) using a certain ratio that can be pre-calculated based on the existing labels achieved from the conventional optimization 2) using either $\mathbf{W}$ or $\mathbf{Q}$ as provided by the DNN and adjust the other one to satisfy the constraint. We tried both and concluded that the latter provides superior performance. In particular we adapt $\mathbf{Q}$ as provided by DNN and use formula (4.9) to scale $\mathbf{W}$. The post processing step and the overall semi-supervised approach is illustrated in Figure 5.5.

5.2.6 Complexity Analysis of DNN

The complexity of training a neural network that has n inputs, l hidden layers each one with $m_1, m_2, ..., m_l$ neurons and k outputs with back-propagation algorithm after N_e epochs and N_s samples is $\mathcal{O}(N_\mathrm{e}N_\mathrm{s}(nm_1 + \sum_{i=1}^{l-1} m_i m_{i+1} + m_l k))$. However, in the forward path it is only $\mathcal{O}(nm_1 + \sum_{i=1}^{l-1} m_i m_{i+1} + m_l k)$.

The beauty of the proposed scheme is that it takes an enormous chunk of the computational complexity offline. This enables the forward path to deliver solution

with a low computational complexity compared to online optimization algorithms such as Interior point method which is often used for non-convex optimizations problems. Interior point method has the worst-case computational complexity of $\mathcal{O}(\max\{n,m\}^4\sqrt{n}\log(\frac{1}{\varepsilon}))$, where n is the number of variables, m is the number of constraints, and ε is the solution accuracy [44].

5.3 Numerical Results and Discussion

In this section we present the performance results of the proposed semi-supervised learning approach for solving the CRAN optimization problem. We consider the same scenario in Section 4.6, i.e., specific weather condition and SIC level, is considered to gather data and train the DNN. Also, we used the same data-set gathered in Chapter 4, i.e., for each weather condition 20000 labeled samples including 16000 training samples and 4000 validation samples.

The DNN structure of both methods are listed in Table 5.2 and input and output sizes are shown in Table 5.1. We divided the training data set into 80 batches of size 200 and used batch gradient and RMSprop for training. To get the final sum rate performance results, we used a semi-supervised DNN to obtain $\mathbf{V}$ and an unsupervised DNN to obtain $\mathbf{Q}$ and $\mathbf{W}$ as explained in Sections 5.2.3 and 5.2.4. For testing, we take the two trained DNNs and in forward path generated the optimal $\mathbf{V}$, $\mathbf{Q}$, and $\mathbf{W}$ and pass them to (2.15) to evaluate the sum rate performance. Table 5.3 illustrates the semi-DNN performance and DNN sum-rate of the proposed scheme. Results show that the semi-supervised learning approach for optimizing backhaul capacity, outperforms the conventional analytical approach used in Chapter 3 by 30%.

Overall, the sum-rate achieved by unsupervised method introduced in this chapter outperforms the supervised method (Chapter 4) by 6% and achieves 79% of the analytical method.

Computational Complexity Comparison

Considering the two DNNs introduced in Tables 5.1 and 5.2 , the overall computational complexity of our unsupervised data-driven methods is $\mathcal{O}(80040) + \mathcal{O}(101800) \approx \mathcal{O}(181 \times 10^3)$, which is 2.7 times faster than the supervised in for-

Table 5.1: Dimensionality of inputs and outputs of unsupervised DNNs.

Parameter	Input & Dimension	Output & Dimension
$\mathbf{W}, \mathbf{Q}$	$\mathbf{CSI_{DL}}$: 20000×64	$\mathbf{W}, \mathbf{S}$: 20000×128
$\mathbf{V}$	$\mathbf{CSI_{FL}}$: 20000×160	$\mathbf{A}$: 20000×200

Table 5.2: Architecture of unsupervised DNNs

Variable	Layers & nodes
$\mathbf{W}, \mathbf{Q}$	64, 100, 150, 150, 130, 128
$\mathbf{V}$	160, 200, 200, 100, 50, 96

ward path and 50000 times faster than model-based method. Notice that, gradient computational graph,that Pytorch uses for calculating the gradient, in unsupervised network is more complicated compared to the supervised method. Hence, even though the unsupervised method is faster in forward path, it requires more training time.

Table 5.3: Performance of semi-supervised and unsupervised DNN-based DL CRAN optimization.

Variable	semi-DNN Performance	DNN Sum-rate
$\mathbf{W}, \mathbf{Q}$	80%	79%
$\mathbf{V}$	130%	

5.3.1 Training Challenges

During training and turning, we faced several challenges that will be explained in this section.

Tuning learning rate

The learning rate is considered the most important tuning parameter in the network. A high learning rate causes fast convergence to local minima and oscillating gradient, and a low learning rate lowers the convergence speed considerably. We tried many different learning rates and used Figure 5.6 to find a proper learning rate

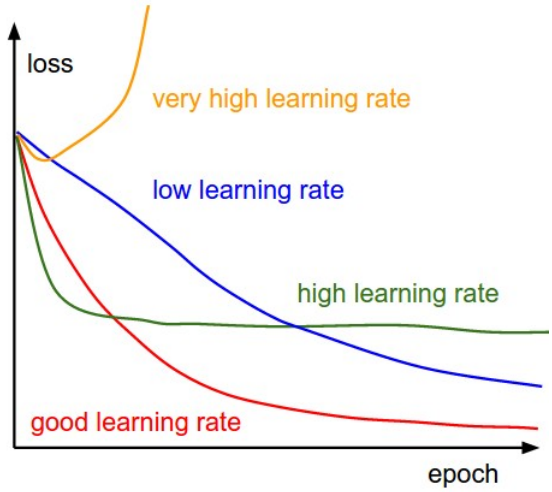

Figure 5.6: Learning rate tuning guide [4]

[4]. Other than the initial learning rate, which is a crucial factor, we can schedule the learning rate to decay in specific intervals (e.g., every 10 epochs) or base on a specific threshold (e.g., train error threshold).

Vanishing Gradient

The activation function is an essential parameter of the network that should be appropriately chosen. For some activation functions such as Sigmoid, after adding several layers to the DNN, the gradients of the loss function approaches zero which stops the model from learning. Easiest and maybe the most practical way to solve the vanishing gradient problem is using Rectified Linear Units (ReLU) or Leaky ReLu activation functions. Also, batch normalization can help with this issue. Another approach is changing the weight initialization from the usual Glorot initialization to He initialization. It adds to network learning ability and has been shown that works well with the Relu activation function [67]. We used all three workarounds in our model and effectively solved the vanishing gradient problem.

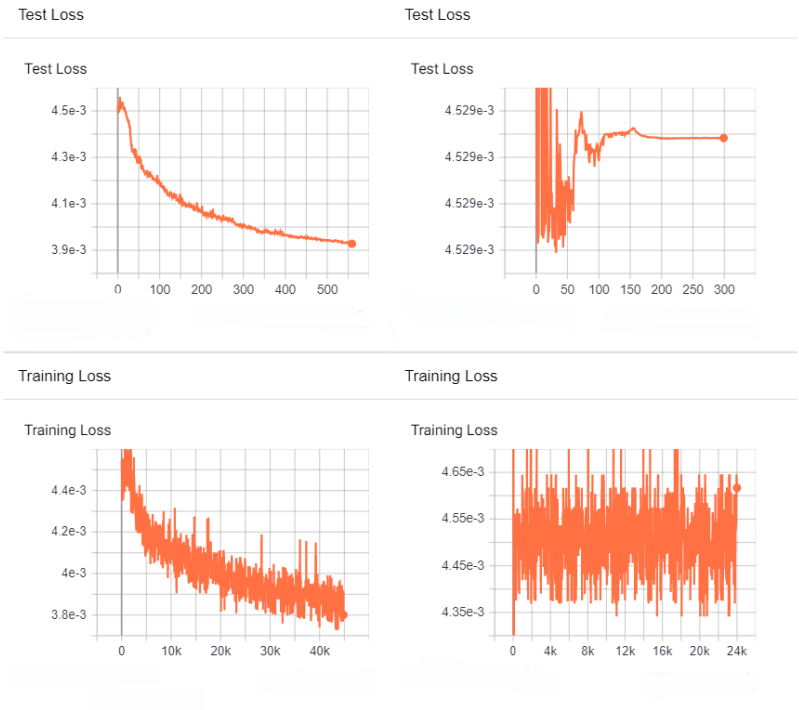

Figure 5.7: Exploding Gradient Illustration. The graph on the right-hand-side shows exploding gradient effect and instabilities in train and test errors, we were able to limit the gradient value by clipping, as it is shown in the left graph.

Exploding Gradient

The exploding gradient is another problem caused when large gradient results in large steps during the weight and biases update and instabilities the optimization, as illustrated in Figure 5.7. One common way to prevent gradient explosion is to use gradient clipping. It sets a limit on the norm of the gradient during the training for more stability. We used gradient clipping, as illustrated in Figure 5.7.

Over-fitting

Another common issue in the training phase is over-fitting, which means that the DNN memorizes the training samples. The most apparent clue of an over-fitted network is the increase in validation error while the training error continues to go

down. We avoided over-fitting in our training by employing the dropout scheme and L2-norm regularization.

Non-homogeneous Loss and Constraints

As mentioned before, in semi-supervised learning, our goal is to minimize the customized loss function formulated in section 5.2.2. The idea is to penalize performance rate for predictions outside of the feasible set. The challenge here is that the value of $\Omega_{\mathbf{V}}$ and the fornthaul capacity which is the objective for optimizing $\mathbf{V}$ and the value of Ω and the user sum-rate which is the objective for optimizing $\mathbf{Q}$ and $\mathbf{W}$ are not homogeneous. So tuning η for adjusting the values is very important during training. For best results, we monitored the sum-rate and penalty values separately during the training phase (same as what we did for the train and validation loss values), and re-tuned η, every couple of epochs, according to the loss, performance and penalty values.

5.3.2 PyTorch Limitations

Despite all the benefits, available libraries and descriptive documentation of PyTorch, there were some challenges we faced working with complex numbers in PyTorch environment. As we discussed in Section 4.6.1, PyTorch automatically creates a computational graph and calculates the gradient for every operation. However, this is true if the function we are using has a defined gradient in PyTorch library. Unfortunately, currently there is a limited algebra support in PyTorch for complex numbers [68, 69] and even some of them are limited to CPU usage only. It made us write many customized functions for basic algebraic operations and restricted our actions to a limited set of operators. It also affected the computation speed as having customized function expands the computational graph. Fortunately, PyTorch is working on expanding the complex library and hopefully they will provide a more straight-forward program development environment for working with complex numbers.

Chapter 6

Conclusion and Future Work

To conclude this thesis, we provide a summary of our main motivations, achievements, and contributions in solving the downlink CRAN optimization problem. We also, propose possible future research directions that reflect the potential openings in this area.

6.1 Conclusion

In this research, we consider the downlink of a cloud radio access network consisting of a central processor and a network of connected radio units. CRAN offers a promising solution for 5G cellular networks as it enables coordinated beamforming and better interference management in ultra-dense networks, which will be one of the most common scenarios in 5G. To address CRAN's backhaul capacity limitation, we proposed a novel resource allocation solution for the scenario with full-duplex self-backhauling RUs connected through hybrid RF/FSO links to the CP for improved network throughput. We studied the feasibility of the IBFD communication in terms of the required self-interference cancellation to outperform the benchmark half-duplex hybrid RF/FSO transmission. Since the derived optimization problem for the design of the linear precoders and quantizers subject to the fronthaul capacity, zero-forcing, and power constraints, is non-convex and intractable, we developed an algorithm to solve it via an alternating optimization approach. In the simulation results, the proposed hybrid RF/FSO system is

assessed in terms of achievable rate, and we highlighted the parameter range for which FD transmission is more rewarding than the time-division approach under different weather conditions and selected RF bandwidth.

Notwithstanding the benefits of the proposed solution, which is based on a conventional optimization approach, this method's computational complexity is still a challenge in practical scenarios. That led us to use a data-driven strategy to reduce complexity. We employed two machine learning-based optimization approaches—supervised and unsupervised to optimize the design variables without solving conventional analytical optimization tasks.

The main achievements of the work in this thesis can be summarized as below:

- We integrated in-band full-duplex self-backhauling and hybrid RF/FSO backhaul in a CRAN architecture to improve backhaul reliability and facilitate coordinated beamforming for better interference management in 5G. We showed full-duplex self-backhauling is a more efficient approach compared to the state-of-the-art half-duplex approach provided enough SIC.

- To address the complexity crunch in the model-based CRAN design, we proposed two data-driven approaches for reducing the computational complexity. We proposed a supervised and an unsupervised learning-based optimization approaches and showed that they have much lower computational complexity compared to the model-based method at the cost of a benign performance loss.

- The key challenge in data-driven optimization approaches was generating solutions that meet all constraints. It required particular DNN architecture, customized activation, and loss functions and post-processing actions; in the case of supervised learning, we imposed constraints via several post-processing steps and proposed a novel DNN architecture to meet positive semi-definiteness constraint. In unsupervised learning, we proposed two activation functions for zero-forcing and positive semi-definiteness constraints, and presented a piece-wise regularization approach that enforces the DNN to generate solutions in the feasible set.

6.2 Future Work

In this section, we discuss possible future research directions.

- Inaccurate channel state information can affect the system's performance and generate a gap between theoretical analysis and real implementation [70], [71]. In this research, we assumed perfect CSI is available at the CP. For future work, we can extend the work to a more realistic imperfect CSI. In addition, we can also investigate the CSI measurement methods and employ the advanced ML-based solutions for channel estimation as well [72]. Also, the channel model of the FSO link can be further enhanced using deep learning frameworks as proposed by [73].

- Another active area of research in 5G is millimeter wave communication. CRAN in such high frequencies needs to employ massive MIMO to combat propagation loss which further complicates the optimization problem in terms of computational cost. The DNN approach proposed in this study can potentially also be applied to millimeter wave CRAN.

Bibliography

[1] B. He and R. Schober. Bit-interleaved coded modulation for hybrid RF/FSO systems. *IEEE Trans. Commun.*, 57(12):3753–3763, 2009. → pages x, 23

[2] 2008-2020 Compute Canada. Cedar - cc doc. → pages x, 32

[3] Dinesh Bharadia, Emily McMilin, and Sachin Katti. Full duplex radios. In *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, pages 375–386, 2013. → pages xi, 12, 13, 14

[4] https://cs231n.github.io/neural-networks-3/. → pages xii, 59

[5] et al. Jonsson, P. Ericsson mobilityreport. *Ericsson: Stockholm, Sweden*, 2020. → pages 1

[6] Tony QS Quek, Mugen Peng, Osvaldo Simeone, and Wei Yu. *Cloud radio access networks: Principles, technologies, and applications*. Cambridge University Press, 2017. → pages 2, 3

[7] D. Gesbert, S. Hanly, H. Huang, S. Shamai Shitz, O. Simeone, and W. Yu. Multi-cell mimo cooperative networks: A new look at interference. *IEEE Journal on Selected Areas in Communications*, 28(9):1380–1408, 2010. → pages 2

[8] A. Mostafa and L. Lampe. Downlink optimization in cloud radio access networks with hybrid RF/FSO fronthaul. In *IEEE Globecom Workshops (GC Wkshps)*, pages 1–7, 2018. → pages 2, 4, 9, 13, 15, 16, 18, 20, 21, 22, 23

[9] J. Lee, Y. Kim, H. Lee, B. L. Ng, D. Mazzarese, J. Liu, W. Xiao, and Y. Zhou. Coordinated multipoint transmission and reception in lte-advanced systems. *IEEE Communications Magazine*, 50(11):44–50, 2012. → pages 3

[10] W. Lee, I. Lee, J. Sam Kwak, B. Ihm, and S. Han. Multi-bs mimo cooperation: challenges and practical solutions in 4g systems. *IEEE Wireless Communications*, 19(1):89–96, 2012. → pages 3

[11] S. Chia, M. Gasparroni, and P. Brick. The next challenge for cellular networks: backhaul. *IEEE Microwave Magazine*, 10(5):54–66, 2009. → pages 3, 4

[12] B. Dai and W. Yu. Sparse beamforming and user-centric clustering for downlink cloud radio access network. *IEEE Access*, 2:1326–1339, 2014. → pages 3, 4

[13] S. Park, O. Simeone, O. Sahin, and S. Shamai. Joint precoding and multivariate backhaul compression for the downlink of cloud radio access networks. *IEEE Transactions on Signal Processing*, 61(22):5646–5658, 2013. → pages 4, 11, 17

[14] P. Patil and W. Yu. Hybrid compression and message-sharing strategy for the downlink cloud radio-access network. In *2014 Information Theory and Applications Workshop (ITA)*, pages 1–6, 2014. → pages 3

[15] P. Patil, B. Dai, and W. Yu. Performance comparison of data-sharing and compression strategies for cloud radio access networks. *European Signal Processing Conference*, pages 2456–2460, 2015. → pages 4, 8

[16] F. Demers, H. Yanikomeroglu, and M. St-Hilaire. A survey of opportunities for free space optics in next generation cellular networks. In *2011 Ninth Annual Communication Networks and Services Research Conference*, pages 210–216, 2011. → pages 4

[17] S. Kazemlou, S. Hranilovic, and S. Kumar. All-optical multihop free-space optical communication systems. *Journal of Lightwave Technology*, 29(18):2663–2669, 2011. → pages 4

[18] M. Z. Chowdhury, M. K. Hasan, M. Shahjalal, M. T. Hossan, and Y. M. Jang. Optical wireless hybrid networks: Trends, opportunities, challenges, and research directions. *IEEE Communications Surveys Tutorials*, 22(2):930–966, 2020. → pages 4

[19] D. Kim, H. Lee, and D. Hong. A survey of in-band full-duplex transmission: From the perspective of phy and mac layers. *IEEE Communications Surveys Tutorials*, 17(4):2017–2046, 2015. → pages 4

[20] S. Hong, J. Brand, J. I. Choi, M. Jain, J. Mehlman, S. Katti, and P. Levis. Applications of self-interference cancellation in 5g and beyond. *IEEE Communications Magazine*, 52(2):114–121, 2014. → pages 4, 7

[21] M. A. Khalighi and M. Uysal. Survey on free space optical communication: A communication theory perspective. *IEEE Communications Surveys Tutorials*, 16(4):2231–2258, 2014. → pages 6

[22] Isaac I Kim and Eric J Korevaar. Availability of free-space optics (fso) and hybrid fso/rf systems. In *Optical Wireless Communications IV*, volume 4530, pages 84–95. International Society for Optics and Photonics, 2001. → pages 6

[23] A. S. Acampora and S. V. Krishnamurthy. A broadband wireless access network based on mesh-connected free-space optical links. *IEEE Personal Communications*, 6(5):62–65, 1999. → pages 6

[24] H. Lei, Z. Dai, K. Park, W. Lei, G. Pan, and M. Alouini. Secrecy outage analysis of mixed rf-fso downlink swipt systems. *IEEE Transactions on Communications*, 66(12):6384–6395, 2018. → pages 6

[25] A. Sabharwal, P. Schniter, D. Guo, D. W. Bliss, S. Rangarajan, and R. Wichman. In-band full-duplex wireless: Challenges and opportunities. *IEEE Journal on Selected Areas in Communications*, 32(9):1637–1652, 2014. → pages 7

[26] Z. Zhang, K. Long, A. V. Vasilakos, and L. Hanzo. Full-duplex wireless communications: Challenges, solutions, and future research directions. *Proceedings of the IEEE*, 104(7):1369–1409, 2016. → pages 7, 23

[27] L. Zhang, M. Xiao, G. Wu, M. Alam, Y. Liang, and S. Li. A survey of advanced techniques for spectrum sharing in 5G networks. *IEEE Wireless Commun.*, 24(5):44–51, 2017. → pages

[28] Dinesh Bharadia and Sachin Katti. Full duplex MIMO radios. In *Symposium on Networked Systems Design and Implementation*, pages 359–372, 2014. → pages

[29] R. Pitaval, O. Tirkkonen, R. Wichman, K. Pajukoski, E. Lahetkangas, and E. Tiirola. Full-duplex self-backhauling for small-cell 5G networks. *IEEE Wireless Commun.*, 22(5):83–89, 2015. → pages

[30] T. Chen, M. Baharaani, J. Zhou, H. Krishnaswamy, and G. Zussman. Wideband full-duplex wireless via frequency-domain equalization: Design and experimentation. In *Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019. → pages

[31] D. Korpi, T. Riihonen, A. Sabharwal, and M. Valkama. Transmit power optimization and feasibility analysis of self-backhauling full-duplex radio access systems. *IEEE Transactions on Wireless Communications*, 17(6):4219–4236, 2018. → pages 7

[32] A. Zappone, M. Di Renzo, M. Debbah, T. T. Lam, and X. Qian. Model-aided wireless artificial intelligence: Embedding expert knowledge in deep neural networks for wireless system optimization. *IEEE Vehicular Technology Magazine*, 14(3):60–69, 2019. → pages 8

[33] H. Ye, G. Y. Li, and B. Juang. Power of deep learning for channel estimation and signal detection in ofdm systems. *IEEE Wireless Communications Letters*, 7(1):114–117, 2018. → pages 8

[34] S. P. Herath and T. Le-Ngoc. Sum-rate performance and impact of self-interference cancellation on full-duplex wireless systems. In *IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pages 881–885, 2013. → pages 9, 15, 21

[35] A Goldsmith. *Wireless communications*. Cambridge university press, 2005. → pages 12

[36] W. Zhang, S. Hranilovic, and C. Shi. Soft-switching hybrid fso/rf links using short-length raptor codes: Design and implementation. *IEEE Journal on Selected Areas in Communications*, 27(9):1698–1708, 2009. → pages 13, 16

[37] A. Molisch. *Wireless communications*, volume 34. John Wiley & Sons, 2012. → pages 14, 15

[38] H. Tabassum, A. H. Sakr, and E. Hossain. Analysis of massive mimo-enabled downlink wireless backhauling for full-duplex small cells. *IEEE Transactions on Communications*, 64(6):2354–2369, 2016. → pages 15

[39] M. Duarte, C. Dick, and A. Sabharwal. Experiment-driven characterization of full-duplex wireless systems. *IEEE Transactions on Wireless Communications*, 11(12):4296–4307, 2012. → pages 16

[40] M. Jain, J. Choi, T. Kim, D. Bharadia, S. Seth, K. Srinivasan, PH. Levis, S. Katti, and P. Sinha. Practical, real-time, full duplex wireless. In *Annual International Conference on Mobile Computing and Networking*, pages 301–312, 2011. → pages 16

[41] Q. H. Spencer, A. L. Swindlehurst, and M. Haardt. Zero-forcing methods for downlink spatial multiplexing in multiuser mimo channels. *IEEE Transactions on Signal Processing*, 52(2):461–471, 2004. → pages 16, 52

[42] Q. Shi, M. Razaviyayn, Z. Luo, and C. He. An iteratively weighted mmse approach to distributed sum-utility maximization for a mimo interfering broadcast channel. *IEEE Transactions on Signal Processing*, 59(9):4331–4340, 2011. → pages 18, 19

[43] K. Akcapinar and O. Gurbuz. Full-duplex bidirectional communication under self-interference. In *International Conference on Telecommunications (ConTEL)*, pages 1–7, 2015. → pages 21

[44] Z. Luo, W. Ma, A. M. So, Y. Ye, and S. Zhang. Semidefinite relaxation of quadratic optimization problems. *IEEE Signal Processing Magazine*, 27(3):20–34, 2010. → pages 22, 57

[45] Ke Li and Jitendra Malik. Learning to optimize. *arXiv preprint arXiv:1606.01885*, 2016. → pages 28, 47

[46] Q. Cappart, T. Moisan, L. Rousseau, I . Prémont-Schwarz, and A. Cire. Combining reinforcement learning and constraint programming for combinatorial optimization. *arXiv preprint arXiv:2006.01610*, 2020. → pages 47

[47] V. Miagkikh and W. Punch. An approach to solving combinatorial optimization problems using a population of reinforcement learning agents. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation-Volume 2*, pages 1358–1365, 1999. → pages 28, 47

[48] G. C. Sobabe, Y. Song, X. Bai, and B. Guo. A cooperative spectrum sensing algorithm based on unsupervised learning. In *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pages 1–6, 2017. → pages 28

[49] W. Song, F. Zeng, J. Hu, Z. Wang, and X. Mao. An unsupervised-learning-based method for multi-hop wireless broadcast relay selection in urban vehicular networks. In *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, pages 1–5, 2017. → pages 28

[50] M. S. Parwez, D. B. Rawat, and M. Garuba. Big data analytics for user-activity analysis and user-anomaly detection in mobile wireless network. *IEEE Transactions on Industrial Informatics*, 13(4):2058–2065, 2017. → pages 28

[51] L. Wang and S. Cheng. Data-driven resource management for ultra-dense small cells: An affinity propagation clustering approach. *IEEE Transactions on Network Science and Engineering*, 6(3):267–279, 2019. → pages 28

[52] W. Cui, K. Shen, and W. Yu. Spatial deep learning for wireless scheduling. *IEEE Journal on Selected Areas in Communications*, 37(6):1248–1261, June 2019. → pages 28

[53] H. Huang, W. Xia, J. Xiong, J. Yang, G. Zheng, and X. Zhu. Unsupervised learning-based fast beamforming design for downlink mimo. *IEEE Access*, 7:7599–7605, 2019. → pages 28

[54] Y. Yang, F. Gao, X. Ma, and S. Zhang. Deep learning-based channel estimation for doubly selective fading channels. *IEEE Access*, 7:36579–36589, 2019. → pages 28

[55] C. Chun, J. Kang, and I. Kim. Deep learning-based channel estimation for massive mimo systems. *IEEE Wireless Communications Letters*, 8(4):1228–1231, 2019. → pages 28

[56] M. Schmidt. Cpsc 340 and 532m - machine learning and data mining (fall 2019). → pages 30

[57] MATLAB. Execute for-loop iterations in parallel on workers - matlab parfor. → pages 32

[58] Junbeom Kim, Hoon Lee, Seung-Eun Hong, and Seok-Hwan Park. Deep learning methods for universal miso beamforming. *IEEE Wireless Communications Letters*, 2020. → pages 33

[59] Mehran Soltani, Vahid Pourahmadi, Ali Mirzaei, and Hamid Sheikhzadeh. Deep learning-based channel estimation. *IEEE Communications Letters*, 23(4):652–655, 2019. → pages

[60] Foad Sohrabi, Kareem M Attiah, and Wei Yu. Deep learning for distributed channel feedback and multiuser precoding in fdd massive mimo. *arXiv preprint arXiv:2007.06512*, 2020. → pages 33, 39, 48

[61] C. Trabelsi, O. Bilaniuk, Y. Zhang, D. Serdyuk, S. Subramanian, J. Felipe Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J Pal. Deep complex networks. *arXiv preprint arXiv:1705.09792*. → pages 34

[62] NumPy®. https://numpy.org/doc/stable/reference/generated/numpy.reshape.html. → pages 34

[63] T .Tieleman and G. Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31, 2012. → pages 42

[64] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016. → pages 47

[65] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos. Learning to optimize: Training deep neural networks for interference management. *IEEE Transactions on Signal Processing*, 66(20):5438–5453, 2018. → pages

[66] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. Sidiropoulos. Learning to optimize: Training deep neural networks for wireless resource management. In *2017 IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pages 1–6. IEEE, 2017. → pages 47

[67] S. Kumar. On weight initialization in deep neural networks. *arXiv preprint arXiv:1704.08863*, 2017. → pages 59

[68] Torch Contributors. Copyright 2019. torch.utils.data — pytorch 1.7.0 documentation. → pages 61

[69] PyTorch. Complex numbers — pytorch 1.7.0 documentation. → pages 61

[70] A. Tajer, N. Prasad, and X. Wang. Robust linear precoder design for multi-cell downlink transmission. *IEEE Transactions on Signal Processing*, 59(1):235–251, 2011. → pages 64

[71] R. Apelfröjd and M. Sternad. Robust linear precoder for coordinated multipoint joint transmission under limited backhaul with imperfect csi. In *2014 11th International Symposium on Wireless Communications Systems (ISWCS)*, pages 138–143, 2014. → pages 64

[72] P. Dong, H. Zhang, G. Ye Li, I. Sim˜oes Gaspar, and N. NaderiAlizadeh. Deep cnn-based channel estimation for mmwave massive mimo systems. *IEEE Journal of Selected Topics in Signal Processing*, 13(5):989–1000, 2019. → pages 64

[73] H. Lee, S. Hyun Lee, T. QS Quek, and I Lee. Deep learning framework for wireless systems: Applications to optical wireless communications. *IEEE Communications Magazine*, 57(3):35–41, 2019. → pages 64