Quantization of Neural Network Equalizers in Optical Fiber Transmission Experiments

Jamal Darweesh

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAT025

Thèse de doctorat

Quantization of Neural Network Equalizers in Optical Fiber Transmission Experiments

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom Paris

École doctorale n°626 de l'Institut Polytechnique de Paris (ED IP Paris)
Spécialité de doctorat : Réseaux, informations et communications

Thèse présentée et soutenue à Palaiseau, le 12/09/2023, par

JAMAL DARWEESH

Composition du Jury :

Pascal BESNARD Professeur, ENSSAT, Lannion (Foton)	Président / Examinateur
Vincent CHOQUEUSE Maitre de Conférence, ENIB, Brest	Rapporteur
Christelle AUPETIT-BERTHELEMOT Professeur, Université de Limoges, Limoges (ENSIL-ENSCI)	Rapportrice
Phillippe CIBLAT Professeur, Télécom Paris, Palaiseau (LTCI)	Examinateur
Mansoor YOUSEFI Maitre de Conférence, Télécom Paris, Palaiseau (LTCI)	Directeur de thèse
Yves JAOUEN Professeur, Télécom Paris, Palaiseau (LTCI)	Co-directeur de thèse
Nelson COSTA Infinera, Portugal	Invité

To my Family, Your unwavering support has been my rock throughout this journey. This manuscript is a tribute to your love and encouragement. With gratitude, JAMAL

Acknowledgements

I would like to express my deepest gratitude to the following individuals and groups who have been instrumental in my successful completion of this PhD thesis.

First and foremost, I am immensely thankful to my academic advisors, for there unwavering support, guidance, and expertise. Your mentorship has been invaluable, and I have grown as a researcher under your tutelage.

I would also like to extend my appreciation to my industrial advisors for their insightful feedback and constructive criticism, which significantly contributed to the quality of this work.

To the members of my thesis committee, I am grateful for your time and expertise in evaluating my research and offering valuable suggestions for improvement.

My heartfelt thanks go to my family for their unwavering support and encouragement throughout this journey. Your love and belief in me were the driving forces behind my perseverance. I would also like to thank my friends, who provided moral support, shared in my triumphs and challenges, and made this journey an enriching experience.

The completion of this thesis was made possible through the financial assistance provided by the European Commission, as this PhD is within the framework of the European Union's Horizon 2020 MSCA-ITN-EID REAL-NET project, under the grant agreement number 81314.

Abstract

The advent of coherent detection marked a significant breakthrough in the field of optical communication. It opened the door to a new era of compensating for the effects experienced by optical signals during their transmission through fiber optic networks. This approach harnessed the power of digital signal processing (DSP) to address the challenges in the electrical domain.

The interplay among chromatic dispersion (CD), Kerr nonlinearity, and Amplified Spontaneous Emission (ASE) noise imposes constraints on the potential capacity of optical fiber systems. As the demand for data transmission within optical networks continues to surge, addressing these effects has risen to the forefront as a significant and pressing research challenge. In this era of expanding optical traffic, the imperative to mitigate these factors has become more pronounced than ever, necessitating innovative solutions to unlock the full potential of optical fiber technology

The classical digital coherent receiver has shown to effectively mitigate linear effects such as CD and polarization mode dispersion (PMD). However the compensation of the nonlinear distortions remains challenging.

Traditional techniques like Digital Back-Propagation (DBP) have been effective in mitigating the deterministic effects arising from the fiber nonlinearity, but come at the expense of increased complexity. DBP necessitates accurate knowledge of the fiber link parameters, making it intricate to implement in practical systems. In this context, the pursuit of low-complexity solutions for addressing nonlinear distortions in optical fiber communication remains a significant endeavor, as it can greatly enhance the practicality and efficiency of optical networks.

In this work, we consider neural networks (NNs) for nonlinearity mitigation in dual polarization optical fiber transmission. Compared to the DBP, NNs do not require the fiber link parameters, and may mitigate the impairments with lower complexity.

We propose two low-complexity NN equalizers: a convolutional-dense and an LSTM-dense model, placed at the end of the linear DSP to compensate the nonlinearities. These equalizers are evaluated in the context of three dual-polarization transmission experiments: a 9x50km true-wave classic fiber link, a 9x110km standard single-mode fiber link, and a 17x70km LEAF fiber link. It is shown that the proposed NNs and DBP achieve about the same Q-factors, both outperforming the linear DSP.

We use quantization in order to reduce the computational complexity, storage size

and energy consumption of the NN equalizers. We compare a number of post-training quantization (PTQ) and training-aware quantization (TAQ) algorithms for casting the weights and activations of the NN in few bits. For quantization above 5 bits, we show that TAQ with straight-through estimation (STE) outperforms PTQ, since it mitigates the quantization noise during the training to some extent. For a Q-factor drop of less than 0.5 dB compared to the unquantized NN, the storage and computational complexity of the NN can be typically reduced by over 90%. However, there is a bit width cut-off value of around 5 bits below which TAQ fails to outperform the linear DSP. This is because, the approximation of the derivative of the quantizer in the STE is not sufficiently accurate at low bit widths. Further, the proposed low-complexity models are not overparameterized, so that the quantization noise can be mitigated during the training at low bit widths. It is shown that the quantization of the activations has a greater impact on the performance compared to the quantization of the weights.

Finally, we study extreme quantization of the NN equalizers below 5 bits. For this case, we propose three novel algorithms: successive PTQ (SPTQ), alpha-blending (AB) and successive AB (SAB) which is a hybrid algorithm that combines the SPTQ with AB. These algorithms are iterative, and incorporate ideas from PTQ and TAQ. We demonstrate that the weights of the NN can be quantized up to one bit, if the activations are not quantized. Further, it is shown that both weights and activations can be quantized at 2–3 bits, while still notably outperforming the linear equalization. Furthermore, we quantify the impact of the quantization noise arising separately from the weights and activations on the Q-factor performance of the model. The results demonstrate for the first time that low-complexity binary NNs can mitigate nonlinearities in optical fiber communication.

This PhD thesis is in the frame of a European Union's Horizon 2020 MSCA-ITN-EID REAL-NET project, grant agreement no. 813144, in collaboration with Infinera in Germany and Portugal.

Résumé

L'avènement de la détection cohérente a marqué une percée significative dans le domaine de la communication optique. Il a ouvert la voie à une nouvelle ère de compensation des effets subis par les signaux optiques lors de leur transmission à travers les réseaux de fibres optiques. Cette approche a exploité la puissance du traitement numérique du signal (DSP) pour relever les défis dans le domaine électrique.

L'interaction entre la dispersion chromatique (CD), la non-linéarité de Kerr et le bruit d'émission spontanée amplifié (ASE) impose des contraintes sur la capacité potentielle des systèmes de fibres optiques. Alors que la demande de transmission de données au sein des réseaux optiques ne cesse de croître, la prise en compte de ces effets est devenue un défi de recherche majeur et pressant. Dans cette ère de croissance du trafic optique, l'impératif d'atténuer ces facteurs est devenu plus prononcé que jamais, exigeant des solutions innovantes pour libérer tout le potentiel de la technologie des fibres optiques.

Le récepteur cohérent numérique classique a montré son efficacité pour atténuer les effets linéaires tels que la CD et la dispersion des modes de polarisation (PMD). Cependant, la compensation des distorsions non linéaires demeure un défi.

Les techniques traditionnelles telles que la Rétropropagation Numérique (DBP) se sont révélées efficaces pour atténuer les effets déterministes résultant de la non-linéarité de la fibre, mais au prix d'une complexité accrue. La DBP exige une connaissance précise des paramètres de la liaison par fibre, ce qui la rend difficile à mettre en œuvre dans des systèmes pratiques. Dans ce contexte, la recherche de solutions à faible complexité pour aborder les distorsions non linéaires dans la communication par fibre optique demeure un effort significatif, car cela peut considérablement améliorer la praticité et l'efficacité des réseaux optiques.

Dans ce travail, nous considérons les réseaux neuronaux (NN) pour atténuer la non-linéarité dans la transmission de fibres optiques à double polarisation. Comparés à la DBP, les NN ne nécessitent pas les paramètres de la liaison par fibre et peuvent atténuer les altérations avec une complexité moindre.

Nous proposons deux égaliseurs NN à faible complexité : un modèle convolutionnel-dense et un modèle LSTM-dense, placés à la fin du DSP linéaire pour compenser les non-linéarités. Ces égaliseurs sont évalués dans le contexte de trois expériences de transmission à double polarisation : une liaison en fibre classique de 9x50 km, une liaison en fibre monomode standard de 9x110 km et une liaison en fibre LEAF de 17x70 km. Il est

démontré que les NN proposés et la DBP atteignent des facteurs Q similaires, dépassant tous deux le DSP linéaire.

Nous utilisons la quantification pour réduire la complexité de calcul, la taille du stockage et la consommation d'énergie des égaliseurs NN. Nous comparons plusieurs algorithmes de quantification après l'entraînement (PTQ) et de quantification consciente de l'entraînement (TAQ) pour réduire le nombre de bits utilisés pour les poids et les activations du NN. Pour une quantification supérieure à 5 bits, il est démontré que le TAQ avec estimation directe (STE) surpasse le PTQ, car il atténue dans une certaine mesure le bruit de quantification pendant l'entraînement. Pour une diminution du facteur Q de moins de 0,5 dB par rapport au NN non quantifié, la taille du stockage et la complexité de calcul du NN peuvent être généralement réduites de plus de 90%. Cependant, il existe une valeur de coupure de la largeur des bits d'environ 5 bits en dessous de laquelle le TAQ échoue à surpasser le DSP linéaire. Cela est dû au fait que l'approximation de la dérivée du quantificateur dans le STE n'est pas suffisamment précise à de faibles largeurs de bits. De plus, les modèles à faible complexité proposés ne sont pas surparamétrés, de sorte que le bruit de quantification peut être atténué pendant l'entraînement à faible largeur de bits. Il est démontré que la quantification des activations a un impact plus important sur les performances par rapport à la quantification des poids.

Enfin, nous étudions la quantification extrême des égaliseurs NN en dessous de 5 bits. Dans ce cas, nous proposons trois nouveaux algorithmes : PTQ successif (SPTQ), alpha-blending (AB) et alpha-blending successif (SAB), qui est un algorithme hybride combinant SPTQ avec AB. Ces algorithmes sont itératifs et intègrent des idées de PTQ et de TAQ. Nous démontrons que les poids du NN peuvent être quantifiés jusqu'à un bit, si les activations ne sont pas quantifiées. De plus, il est montré que les poids et les activations peuvent être quantifiés à 2-3 bits, tout en surpassant notablement l'égalisation linéaire. De plus, nous quantifions l'impact du bruit de quantification provenant séparément des poids et des activations sur les performances du facteur Q du modèle.

Les résultats montrent pour la première fois que les NN binaires à faible complexité peuvent atténuer les non-linéarités dans les communications par fibre optique.

Cette thèse de doctorat s'inscrit dans le cadre du projet REAL-NET de l'Union européenne dans le cadre du programme Horizon 2020 MSCA-ITN-EID, accord de subvention n° 813144, en collaboration avec Infinera en Allemagne et au Portugal.

Contents

List of Figures

List of Tables

Acronyms

AB alpha-blending

AI Artificial Intelligence

APoT additive PoT

ASE amplified spontaneous emission

ASIC application specific integrated circuit

AutoML automated machine learning

AWGN additive white Gaussian noise

BER bit error ratio

BiLSTM bi-directional long short-term memory

BO bit-wise operations

CD chromatic dispersion

CFO carrier frequency offset

CM constant modulus

CNLSE coupled nonlinear Schrödinger's equation

CNN convolutional neural network

CPE constant phase estimation

DBP digital back propagation

DGD differential group delay

DM dispersion-managed

DSF dispersion-shifted fiber

DSP digital signal processing

EAM electro-absorption modulator

EDFA erbium-doped fiber amplifiers

ERM empirical risk minimization

FFT fast Fourier transform

FIR finite frequency response

FWM four wave mixing

GPU Graphical Processing Unit

GVD group velocity delay

HMM Hidden Markov Model

IFFT inverse fast Fourier transform

iid independent identically distributed

ISI inter-symbol interference

LDBP learned digital back propagation

LO local oscillator

LSTM long short-term memory

MAC multiply-accumulate

MAP maximum a posteriori probability

MIMO multiple input multiple output

ML maximum likelihood

MLP multi layer perceptron

MLSE maximum likelihood sequence estimation

MMF multi mode fiber

MSE mean square error

MZM Mach Zehnder modulator

NAS neural architecture search

NLSE nonlinear Schrödinger equation

NN neural network

NNs neural networks

NZDSF nonzero dispersion-shifted fiber

OPC optical phase conjugation

PAM pulse amplitude modulation

PCA principal component analysis

PDM polarization-division multiplexing

PMD polarization mode dispersion

PoT power of two

PSD power spectral density

PSK phase-shift keying

PSP principal state polarization

PTQ post-training quantization

QAM quadrature amplitude modulation

QAM quadrature amplitude modulation

RDE radially directed equalizer

RNN recurrent neural network

RRC root raised cosine

RX receiver

SAB successive AB

SGD stochastic gradient descent

SNR signal to noise ratio

SOP state of polarization

SPM self-phase modulation

SPTQ successive PTQ

SSFM split-step Fourier method

SSMF standard single-mode fiber

STE straight-through estimator

tanh tangent hyperbolic

TAQ training-aware quantization

TX transmitter

WDM wavelength-division multiplexing

XPM cross-phase modulation

CHAPTER 1

Introduction

The demand for traffic has increased consistently over time. Multiple sources forecast that this trend will continue in the foreseeable future [112, 22]. Optical communication has played an important role in supporting the global Internet traffic. The achievable information rates in the communication networks have increased exponentially in the past decades, thanks to the advances in the fiber-optics technology and digital communications.

Optical fiber is made of thin strands of glass, allowing transmission of light signals over long distances. Lightwave communication is ideal for the transport of the large amounts of information over long distances. Optical fiber has a much lower loss and higher bandwidth than the electronic media. The wavelength-division multiplexing (WDM) makes it possible to transmit parallel bits streams in different wavelengths of light, substantially increasing the throughput in a single fiber. The amplification in the optical domain using, *e.g.,* the erbium-doped fiber amplifiers (EDFA), eliminates the need for excessive regeneration, significantly extending the reach and the capacity of the optical communication systems.

Digital coherent receivers use advanced modulation formats and digital signal processing (DSP) to compensate for the fiber impairments and increase the spectral efficiency. These systems achieve the transmission rates of around 100 terabit per second (Tbps) over a single fiber of hundreds of km. The coherent receiver offers significant benefits, but they come at the cost of the receiver complexity. Research has focused on reducing the energy consumption, latency and the cost of DSP.

This thesis is dedicated to low-complexity equalization in optical fiber transmission using neural network (NN)s. In the remaining part of this chapter, we provide an overview of the equalization in optical fiber, quantization of the NNs used for nonlinearity mitigation, and the outline of the contributions of the thesis.

1.1 Equalization in Optical Fiber Communication

The interaction between the chromatic dispersion (CD), Kerr nonlinearity and amplified spontaneous emission (ASE) noise limits the capacity of optical fiber. With the growth of the traffic in the optical networks, the mitigation of these effects has become an important research problem.

Thanks to the advances in the DSP, linear transmission effects, such as the CD and polarization mode dispersion (PMD), can be efficiently mitigated in the electrical domain using the digital coherent receivers [100]. Linear equalization has low complexity, and is implemented in the practical coherent transmission systems.

However, the mitigation of the nonlinear effects is challenging. Pulse propagation in optical fiber is modeled by the nonlinear Schrödinger equation (NLSE) [2]. The deterministic effects arising from the fiber Kerr nonlinearity, such as the self-phase modulation (SPM), can be mitigated using the digital back propagation (DBP) based on the split-step Fourier method (SSFM) [61]. However, the computational complexity of the DBP can be high, since it potentially requires a large number of the fast Fourier transform (FFT) operations [35].

Neural networks have recently been studied for equalization in optical fiber transmission [65]. Compared to the model-based equalizers such as DBP, NNs does not require side information about the channel, and may offer low-complexity mitigation of the fiber impairments. Two classes of the NN equalizers have been proposed in the literature. In model-driven approaches, the NN architecture is based on the discretization of the NLSE using the SSFM [14]. In contrast, in model-agnostic approaches, the architecture does not depend on the channel [65]. Examples include multi layer perceptron (MLP), convolutional and recurrent models, as well as their combinations [40, 29].

In this work, we consider nonlinearity mitigation in optical fiber using model-agnostic NNs. We study network quantization, in order to reduce the size of the model.

1.2 Quantization of the Neural Network Equalizers

NNs have achieved the state-of-the-art results in classification and regression in a number of application domains. These networks are often over-parameterized, with a large number of weights and biases. It can be difficult to implement such NNs in applications that require real-time inference, low energy consumption, or run in resource-constrained environments.

As a consequence, research has recently focused on reducing the size of the NNs, in order to improve the latency, memory footprint and energy consumption, while maintaining a good prediction accuracy [45].

One approach to low-complexity NNs is optimizing the architecture for a given task. The hyper-parameters of the NN, such as the number of layers and neurons, as well as the type of layers and other aspects of the model, are usually optimized in practice [56, 60, 57]. Traditionally, one would manually search for suitable architectures and hyper-parameters which is not scalable. New methods such as automated machine learning (AutoML) and the neural architecture search (NAS) find good architectures automatically, while adhering to the constraints on the model size, depth and width [36].

Another approach to reducing the computational complexity and memory usage of the model is pruning and quantization. In pruning, some of the neurons are removed from the model, giving rise to a sparse computational graph. For example, neurons with small sensitivity, *e.g.,* those that have small impact on the loss function or output of the model, can be removed. In some cases, most of the neurons or weights can be pruned with little impact on the model's generalization performance. The challenge is finding a suitable trade-off between the level of sparsity of the model and the prediction accuracy [12].

Finally, in quantization, the weights, biases and activations are represented in fewer bits than the full precision 32 bits, subject to a given prediction accuracy. Quantization can be applied in the training as well as the inference mode. In fact, training in half or mixed precision [46, 50] has been a key enabler of the high-throughput Artificial Intelligence (AI) accelerators. However, training below half precision is challenging, and the majority of research has focused on the quantization in the inference mode.

In this work, we investigate different types of quantization, however, at low number of bits, a hybrid approach combining ideas from the post-training and train-aware quantization is the most successful.

1.3 Research Objectives

This thesis aims to design low-complexity NNs for equalization in dual-polarization optical fiber transmission. The NNs are integrated into the existing digital coherent receivers, primarily to mitigate the fiber nonlinear effects. Since these equalizers should eventually be implemented in application specific integrated circuit (ASIC) in practice, it is important to minimize the size of the NN as much as possible. To do so, we use quantization, drawing

on methods and concepts from other domains such as the computer vision. These methods are adapted and applied to the optical fiber transmission.

We present a number of quantization algorithms for casting the weights and activations of the NN equalizers in few bits, in order to reduce the computational complexity and memory requirements. In particular, we propose a novel hybrid quantization algorithm for low-complexity NN nonlinearity mitigation, with as few as 1–3 bits per weight and activation, and excellent performance in bit error ratio (BER).

The goals of this PhD thesis consist of the following.

- Designing NNs tailored to the nonlinearity mitigation in dual-polarization optical fiber transmission, and determining their gain in Q-factor compared to the linear equalization and DBP, in several transmission experiments (the unquantized case).
- Providing a comprehensive comparison of the several algorithms for the quantization of the NN equalizers, including a number of proposed ones [49]. The comparison is made in terms of the Q-factor, computational complexity and memory requirement, for several values of the launch power and quantization rate (the quantized case).
- Establishing the trade-off between the performance (measured in Q-factor) and complexity (measured in the number of bit-wise operation per detected bit) as a function of the launch power; identifying suitable algorithms for each transmission regime.

1.4 The Outline and Contributions of the Thesis

The remaining part of the thesis is structured into the following chapters.

Chapter II provides a brief overview of the coherent transmission over optical fiber. It begins with a historical review of the optical communication, followed by the basics of the optical transmission with the DSP at the receiver. The chapter then reviews the linear and nonlinear effects in dual-polarization transmission over optical fibers. Finally, the conventional coherent receiver, and the associated DSP chain including the DBP, for mitigating the transmission impairments are presented.

Chapter III provides a brief overview of the NNs. We recall a few concepts from the statistical machine learning, and review the architectures used in the subsequent chapters of this thesis: multi-layer perceptron, convolutional and long short-term memory (LSTM). Next, we discuss the training of the NNs using the gradient descent, including the calculation of the gradient. Lastly, we review some of the applications of the deep NNs, in digital

communications and beyond.

Chapter IV is dedicated to the application of the NNs to the nonlinearity mitigation in optical fiber transmission. We begin with a classification and review of the NN-based equalizers used in the literature in the past few years. We propose two low-complexity NN nonlinear equalizers: a CNN-dense and a bi-directional long short-term memory (BiLSTM)-dense model. These models are easily integrated into the conventional coherent receivers, by placing them at the end of the linear DSP chain. We evaluate the BER of the proposed equalizers in three transmission experiments, and quantify their Q-factor gains over the linear equalization and DBP.

In **Chapter V**, we study quantization methods for reducing the size of the NNs proposed in Chapter IV . Two classes of the quantization algorithms are considered: training-aware and post-training quantization. We compare the uniform and non-uniform quantization, and highlight a companding quantization proposed for the first time in the context of equalization in this thesis. Further, we discuss the implementation of the mixed-precision quantization, where different layers are assigned different bit-widths. We compare the performance of the quantization algorithms in terms of the Q-factor drop relative to the unquantized models, as well as the computational complexity and memory requirements. However, we also acknowledge the limitations of these algorithms and potential trade-offs that must be considered when selecting a quantization method.

Finally, in **Chapter VI**, we present two quantization algorithms for nonlinearity mitigation that are particularity well suited to low number of bits. The results show that our proposed method outperforms the existing quantization algorithm at low bit-widths. Overall, this chapter highlights the importance of the quantization in low-complexity nonlinear equalization in optical fiber transmission.

CHAPTER 2

Digital Optical Fiber Transmission Systems

This chapter provides a brief introduction to the digital optical fiber transmission systems. It begins with a historical overview of optical communication, from the regenerative systems to modern coherent transmission with digital signal processing. The chapter then reviews the characteristics of the optical fiber channel, and the pulse propagation models in dual-polarization transmission. The modulation of information in the digital and optical domains are explained, together with the optical amplification schemes. The chapter then describes the operation of the coherent receiver, and the conversion of the optical to electrical signals. The components of the DSP chain at the receiver for the compensation of the linear fiber transmission effects are presented. Lastly, the chapter discusses the nonlinearity mitigation techniques using DSP, and highlights some of the limitation of the current algorithms.

2.1 Historical Overview

The principle of the total internal reflection dates back to centuries ago, and was demonstrated in the 19th century by the physicist John Tyndall and others. Optical fibers were available in 1960s, but were not considered for data communication since the signal would vanish over a few meters. In 1966, Charles Kao and coworkers demonstrated that the high loss of the glass fiber at the time is not all intrinsic, and arises mostly from the impurities in the glass. They predicted that the attenuation can be reduced to below 20 dB/km, comparable to that of the coaxial cables in repeater distance. The American company Corning soon reported a prototype fiber with this value of loss in 1970, followed by a Japanese group attaining 0.2 dB/km at 1.55 μm in 1979 [3]. This set the stage for the advances in

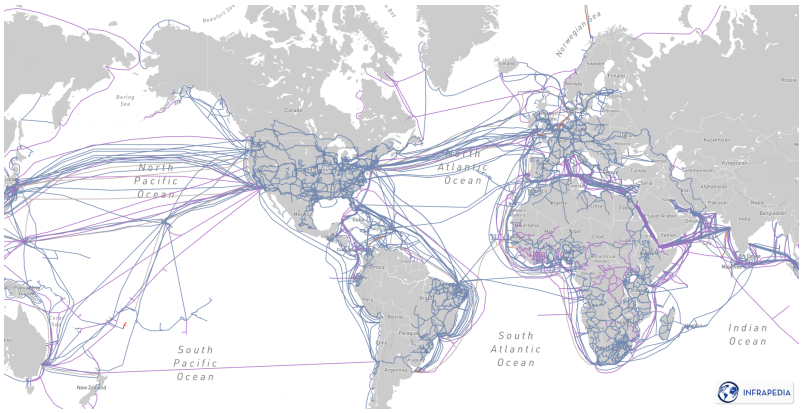


Figure 2.1: Submarine and terrestrial optical fiber cables as of 2023 [59].

fiber-optics that would follow in the decades to come.

The use of optical fiber in communications has since grown tremendously. Today, optical fiber forms the backbone of the telecommunication networks, supporting online services such as steaming, music distribution, social networks, electronics commerce, and arguably the artificial intelligence. The total length of fiber deployed has now surpassed 4 billion kilometers worldwide [119].

2.1.1 The Regenerative Systems

A digital lightwave transmission system successfully communicated signals at the rate of 44.736 Mb/s in field conditions in 1977 [67]. Shortly afterwards, live telephone traffic was successfully sent through multi mode fiber (MMF) by several companies, at the rate of tens of Mb/s [55, 103, 11, 88]. It is worth noting that the type of fiber used in some of the early experiments was MMF, not standard single-mode fiber (SSMF). In the late 1980s, the undersea optical fiber transmission systems installed across the Atlantic and Pacific Ocean were regenerative systems that functioned at the wavelength 1.3 μm [121]. In one deployment, each of the three fiber pairs carried 280 Mb/s of data [121]. The signals needed to be regenerated every tens of km. This repeater distance is larger than 1km in coaxial cables, but still limited the reach of the fiber communication systems.

2.1.2 Erbium-Doped Fiber Amplifiers

The invention of the EDFA [81, 32] allowed for the amplification of optical signals without the need for excessive regeneration. Optical amplification ushered in the era of long-haul fiber transmission. It should be noted that the exponential growth in the capacity of optical links could not have been solely attributed to the invention of the EDFA. Rather,

it was the result of a combination of a number of inventions in photonics components and the effective use of the EDFA bandwidth.

2.1.3 Dispersion Managed WDM Systems

The absence of a practical CD compensation algorithm led to the widespread adoption of the dispersion-shifted fiber (DSF) with near zero dispersion at the operating wavelength, in the early 1990s [87]. However, DSF is prone to the nonlinear distortions, such as the four wave mixing (FWM) which is strong at zero CD [44]. FWM generates new wavelengths that may coherently interfere with the wavelength of interest, and reduce the signal to noise ratio (SNR) [39].

Upon refining the manufacturing process, researchers produced the nonzero dispersion-shifted fiber (NZDSF), which has low but measurable CD [19]. The NZDSF could be produced in two variants, with slightly positive or negative CD values at 1.55 μm. In the dispersion-managed (DM) transmission systems [21], fibers with the opposite dispersion signs are combined to obtain a net CD value of near zero, while maintaining high local CD in distance to mitigate FWM. This technique thus compensates both CD and nonlinearities to some extent, and has been widely used in high-speed dense WDM commercial systems, until the emergence of the digital coherent receivers.

2.1.4 Digital Coherent Receivers

Coherent detection was initially of interest for increasing the distance between the regenerators in the early systems, because it improves the receiver sensitivity compared to the intensity modulation and direct detection (IM/DD) [76]. The technology, however, could not be commercialized due to challenges with the phase and polarization locking. With the success of the EDFA, the span-by-span regenerative systems became outdated, and research on coherent receivers declined.

The authors in [31] showed that the QPSK digital transmission, widely used in radio communication at the time, could also implemented in the optical communication using a digital coherent receiver. The revival of the coherent detection can be attributed to the need to compensate for the distortions caused by CD and PMD, which had been a problem for 40-Gb/s systems in the early 2000s, as well as the advances in the CMOS processing speed. The high-speed CMOS electronics paved the way to the digital electronic dispersion compensation [80], and the implementation of the advanced algorithms such as

the maximum likelihood sequence estimation (MLSE) [4] which was commercially launched at 10 Gb/s [38].

Digital coherent receivers combine the advantages of the homodyne detection with minimal electrical receiver bandwidth, and heterodyne detection with no optical phase locking. Coherent receivers use a local oscillator (LO) laser at receiver (RX) to convert the dual-polarization optical signal to the digital domain. With access to the full optical field digitally, one could leverage advanced modulation formats such as the quadrature amplitude modulation (QAM), and polarization-division multiplexing (PDM), boosting the spectral efficiency by up to 4X.

Coherent transmission increased the data rates in optical communication to 40-Gb/s per channel in mid 2000s, using the 10-Gb/s componentry and CMOS electronics of the time. Moreover, the digital form of the entire optical field opened up the possibility of digital compensation of CD, PMD, optical filtering, and even the fiber nonlinearities.

Digital coherent receivers offer significant benefits, but these advantages come at the cost of the receiver complexity. This includes the necessity of a LO laser at the receiver, and the use of the power-hungry DSP. However, the opto-electronic front-end architecture of the coherent transponders allows for the manufacturing of the components at higher volume compared to that in direct detection. This led to significant investments in the coherent detection technologies. Nortel commercially implemented the first intradyne transponder in 2008 at 40 Gb/s per channel [111]. Alcatel-Lucent followed with a 28 GBaud 100-Gb/s single-wavelength commercial transponder [94]. Over the years, there has been a significant increase in the total deployed fiber, with the majority of it being terrestrial long-distance, followed by submarine, systems.

Digital coherent transponders can compensate for significant amounts of the accumulated dispersion. Research has established that the nonlinear impairments are weaker in the coherent transmission in dispersion-uncompensated links compared to the DM systems. As a result, DM is no longer used in new deployments. Modern single-mode fibers have losses of just over 0.14 dB/km, CD values of approximately 17 ps/(km.nm), and large effective areas to minimize the nonlinear distortions [114, 113].

2.1.5 Capacity Limits of Optical Networks

Optical fibers are known for their remarkable properties such as low loss and large bandwidth. These characteristics led to the belief that optical fibers have almost limitless band-

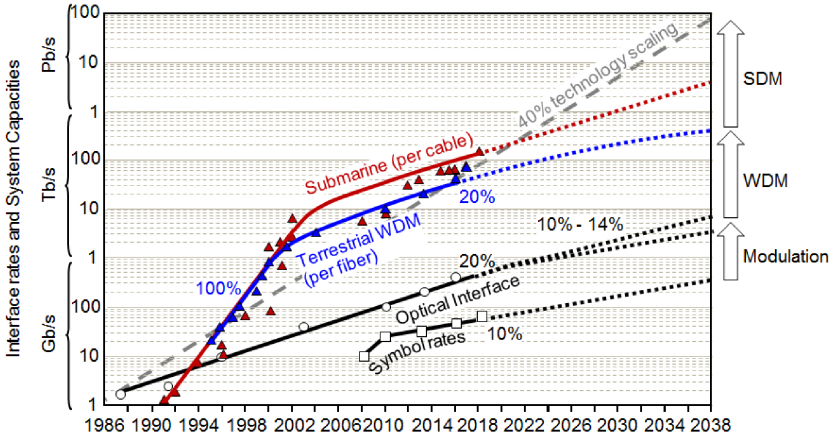


Figure 2.2: Optical transmission capacities over decades [121].

width. Nonetheless, the growth of the achievable information rates slowed down, causing concerns that optical fiber transmission systems might have reached the fundamental limits [120].

The capacity of optical fiber is limited by the interaction between the chromatic dispersion (CD), Kerr nonlinearity and ASE introduced by inline amplifiers [128, 104]. The ASE produces an amplitude noise that is then converted to the phase noise via self-phase modulation [128].

In particular, fiber nonlinearities are a major obstacle in high bit rate transmission systems [86]. Kerr nonlinearity produces a phase shift depending on the signal's intensity. This causes the creation of the new frequency components in the signal spectrum that could act as distortions in WDM.

An information-theoretic technique to determine the capacity of optical fiber was presented in [37]. This approach carefully considered the influence of the Kerr nonlinearity, and deduced a nonlinear Shannon limit. Here, the achievable information rates in optical fiber follow the capacity of the linear Gaussian channels at low powers, but flatten out at high powers due to nonlinear distortions. At the European Conference on Optical Communications in Vienna in 2009, a plenary talk highlighted the implications of nearing the fundamental capacity limits [20].

On the other hand, it has been shown that the capacity $\mathcal{C}$ of optical fiber is upper bounded by the capacity of an additive white Gaussian noise (AWGN) channel with the same SNR, *i.e.,*

$$\mathcal{C} \leq \log_2(1 + \mathrm{SNR}). \tag{2.1}$$

One stochastic effect arising from the signal noise interaction is the Gordon-Mollenauer noise [48]. This is a nonlinear phase noise that particularly impacts the polarization multiplexed systems [48]. However, this effect is significant primarily in transmission systems where the optical pulses undergo minimal amplitude changes, for example, in the zero-dispersion fibers or with soliton transmission [83]. In most cases, the dispersion induces significant pulse broadening, and other impairments become dominant.

The transition from DSF to NZDSF and SSMF shows the relationship between the type of fiber and transponder design, which has evolved since the inception of the optical communication systems. With advances in the transponder technology, new fibers have been continually introduced, subject to the high labor costs of the fiber installment which often exceed all other expenses in the system deployment. Space-division multiplexing (SDM) is a new technology which uses fibers that support multiple spatial modes or cores, in order to transmit parallel data streams. The achievable information rates in fiber scale up with the number of modes or cores, overcoming some of the capacity limitations in optical networks with the single-mode fiber.

2.1.6 Research in Nonlinearity Mitigation

The distortions from the dispersion and nonlinearity are deterministic. Thus, it is theoretically possible to compensate them. One approach is to apply optical phase conjugation (OPC), reversing the phase of the electric field [125, 92]. The could be done using the mid-span spectral inversion, performing phase conjugation at the midpoint of the transmission path [118]. A more efficient approach is mid nonlinearity temporal inversion in the time domain [84, 85].

The DBP was proposed in 2008 to digitally compensate the linear and nonlinear fiber impairments [73, 107]. DBP is based on the SSFM, a numerical algorithm used to simulate the propagation of a pulse through the fiber. [2]. By solving the inverse NLSE, DBP recovers the transmitted signal that has been distorted through propagation.

A dual-polarization DBP algorithm that considers PMD in optical fiber communication was proposed in [24]. By considering the accumulated PMD at the receiver, this algorithm is able to distributively compensate for PMD via reverse propagation, and has been shown to outperform the conventional approaches.

It has been observed that DBP can require significant computational resources due to the need to perform FFT and inverse fast Fourier transform (IFFT) multiple times [61].

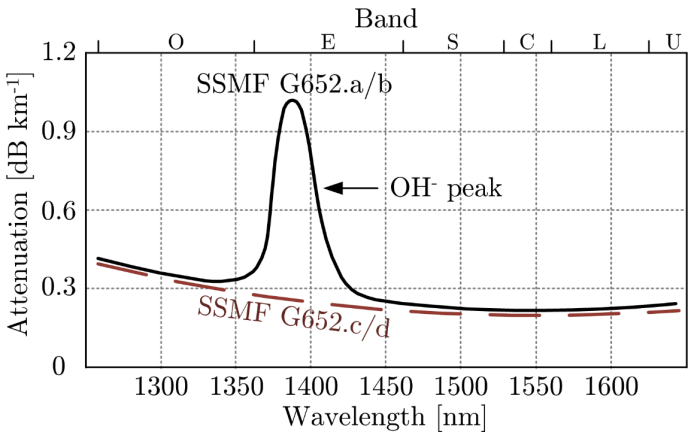


Figure 2.3: Attenuation of the standard single mode fiber in ITU G. 652 [115].

This calls for research in low-complexity nonlinearity mitigation.

We close off this section pointing out that some of the advances in DSP and coding in fiber-optic communications have been incorporated in the industry standards and protocols. For example, the standards such as 100 Gigabit Ethernet (GbE), 400 GbE, and 800 GbE in data centers and networks leverage advanced modulation formats and signal processing algorithms to maximize the information capacity of the optical links.

2.2 Optical Fiber Channel

Optical fiber is designed to efficiently guide the light form the input to destination. The lightwave is nonetheless subject to impairments that accumulate in distance. In this section, we review some of the fiber transmissions effects.

2.2.1 Characteristics of Optical Fiber

Attenuation

Prior to the development of the EDFA, the fiber loss considerably limited the reach of the optical transmission systems. In fact, the signal power decays exponentially with distance: if the power at the input is P_0, the power at the output of a fiber of length L is

$$P_L = P_0 \exp(-\alpha L), \tag{2.2}$$

where α is the loss coefficient.

The loss arises from numerous sources, notably, the Rayleigh scattering and the material absorption. These factors depend on the wavelength, making the loss coefficient α

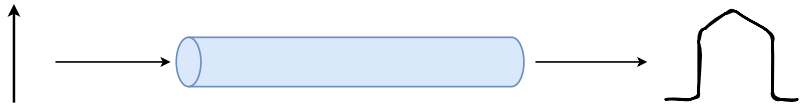


Figure 2.4: Illustration of the effect of the chromatic dispersion on a pulse.

wavelength dependent.

The Rayleigh scattering occurs when light interacts with small particles and scatters in all directions, due to small local variations in the refractive index of the medium. In the silica glass fibers, this scattering is most pronounced at shorter wavelengths [13], where the particles are more likely to interact with the light. In silica, the attenuation coefficient due to the Rayleigh scattering is [105]

$$\alpha_R = \frac{1.89510^{-28}}{\lambda^4} \quad \mathrm{m}^{-1}, \tag{2.3}$$

where λ is the wavelength of light.

The material absorption dissipates some of the transmitted optical power as heat. The absorption can be intrinsic or extrinsic. The intrinsic attenuation occurs due to the interaction of light with the components of the glass and material composition. On the other hand, the extrinsic absorption is caused by impurities in the glass from the fabrication process. This absorption is attributed largely to the water dissolved and integrated into the glass structure. This results in the emergence of harmonics at 1.38, 0.95, and 0.72 μm.

In general, Rayleigh scattering is more significant than the material absorption in optical fibers, especially in the telecommunications band around 1550 nm. However, both phenomena must be considered in the calculation of the overall attenuation coefficient of the fiber. Fig. 2.3 shows the attenuation in dB/km in the SSMF specified in the ITU G. 652 standard [115].

Chromatic Dispersion

Another important effect that changes the communication signals in optical fibers is the chromatic dispersion. The refractive index of the fiber depends on the wavelength. Therefore, different spectral components of the light (colors) travel at different speeds, and arrive at different times at the output. This causes pulse broadening in the time domain.

The effect of the CD can be seen by expanding the propagation constant $\beta(\omega)$ of the

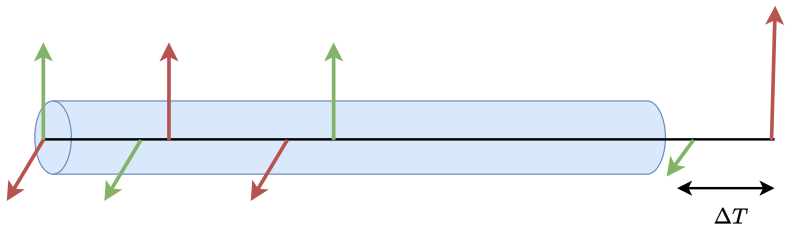


Figure 2.5: The effect of the PMD on the polarized waveform.

signal around the central frequency ω_0

$$\beta(w) = \beta_0 + \beta_1(w - w_0) + \frac{1}{2}\beta_2(w - w_0)^2 + \cdots, \tag{2.4}$$

where

$$\beta_m = \left(\frac{d^m\beta}{dw^m}\right)_{w=w_0}.$$

The zero-order dispersion coefficient β_0 produces a fixed phase shift. The first-order coefficient β_1 reflects the pace at which the pulse envelope advances. The group velocity of the pulse is $v_g = \frac{1}{\beta_1}$. The second-order CD coefficient β_2 determines the group velocity delay (GVD), which characterizes the rate of change of the spectral components of the pulse. The CD is frequently measured by the parameter

$$D = -\frac{2\pi c}{\lambda^2}\beta_2, \tag{2.5}$$

where c is the velocity of the light, and λ is the wavelength.

The pulse broadening caused by CD results in an interaction between the symbols in the time domain, known as the inter-symbol interference (ISI). This restricts the achievable rates and the reach of the optical transmission systems that do not employ CD compensation.

Polarization-Mode Dispersion

Optical fiber can support two orthogonal x and y polarizations. In a perfect fiber, the refractive index along the two axes of the polarizations are equal, *i.e.*, $n_x = n_y$, where n_x and n_y are refractive indices of the x and y polarizations respectively. In this case, the two polarizations travel at the same speed. However, fibers posses certain degree of asymmetry due to imperfections in the manufacturing process or mechanical stress [1]. This asymmetry breaks the degeneracy of the orthogonally polarized modes, resulting in

birefringence that produces a difference in the phase and group velocities of the two modes.

The modal birefringence for the fiber is

$$B = |n_x(\omega_0) - n_y(\omega_0)| \tag{2.6}$$

$$= \frac{\lambda}{2\pi}|\beta_{0x} - \beta_{0y}|, \tag{2.7}$$

where β_{0x} and β_{0y} are the zero-order propagation constants (or dispersion coefficients) of the x and y polarizations respectively, and λ is the optical wavelength.

The difference in the propagation constants of the two polarizations produces a time delay between their signals referred to as the differential group delay (DGD). For constant modal birefringence in distance, at the end of a fiber of length L, the amount of the delay is

$$\Delta T = L|\beta_{1x} - \beta_{1y}|. \tag{2.8}$$

The imperfections and non-idealities in the fiber, and consequently the changes in the axis of the birefringence and DGD, are random. The polarization beat length $L_B = 2\pi/B$ is around 1–10m in SSMF. This implies that the state of polarization (SOP) rotates with a random angle roughly every L_B km along the fiber. Since L_B is much smaller than the typical fiber length, the SOP varies rapidly in distance.

Kerr Nonlinearity

Lightwaves at distinct frequencies traveling in a medium in general do not interact with one another. However, sometimes the transmission medium causes interactions between the propagating waves. These frequency interactions arise from the material nonlinearities.

In optical fibers, the nonlinear effects are usually small. However, they can accumulate as the light travels over hundreds of kilometer. Moreover, the effects become strong at high intensities, or if the signal power is concentrated in a small area, such as the core of a SSMF. The nonlinear phenomena begin to occur at power levels of a few milliwatts in long-haul transmission over SSMF.

The Kerr effect arises in media with a refractive index that depends on the signal intensity. The refractive index is a property of the material that is determined by the induced polarization (not to be confused with the light polarization), namely, how the material responds to an incident electric field. At low intensities, the induced polarization is linearly proportional to the electric field, so that the refractive index does not depend

on the signal. However, at higher optical intensities, this does not hold.

The Kerr effect gives rise to a number of nonlinear effects, the first of which is the SPM. Due to the dependency of the refractive index on the intensity of the light, a phase shift proportional to the intensity is generated. Therefore, when a pulse travels through the fiber, the Kerr effect produces a time-varying phase for the peak of the pulse where the amplitude is large. A time-varying phase in turn produces a time-varying frequency and spectral broadening.

The SPM may limit the transmission rates in long-distance optical communication if not compensated. On the other hand, the authors of [70] demonstrate that this nonlinear phenomenon can also be utilized to compress the duration of an optical pulse, which has useful applications.

In cross-phase modulation (XPM), changes in the intensity of a signal at one wavelength λ_1 will alter the refractive index of the fiber at another wavelength $\lambda_2 \neq \lambda_1$. This causes phase modulation at wavelength λ_2, with an amount proportional to the intensity of the signal at wavelength λ_1. Consequently, XPM is manifest as cross-talk in WDM. As with SPM, the phase modulation causes the frequency modulation and spectral broadening.

The FWM is yet another nonlinear effect caused by the interaction of different frequencies of light. When three frequency components co-propagate in fiber, a wave at a new frequency is generated. This frequency interaction can be especially problematic in WDM transmission, if wavelengths are in close proximity.

2.2.2 Optical Fiber Channel Model

The propagation of a signal in fiber is governed by a balance between the dispersion and nonlinear effects. In this section, we review pulse propagation models in single- and dual-polarization fibers.

Single-polarization model

The propagation of the complex envelope of the signal $q(t, z)$ as a function of time t and distance z in one polarization of the electric field is modeled by

$$\frac{\partial q}{\partial z} = -\frac{\alpha}{2}q - \frac{j\beta_2}{2}\frac{\partial^2 q}{\partial t^2} + j\gamma|q|^2 q, \tag{2.9}$$

where α is the attenuation constant, β_2 is the second-order dispersion coefficient, γ is the nonlinearity parameter, $j = \sqrt{-1}$, and $|.|$ represents the magnitude. The equation (2.9) is

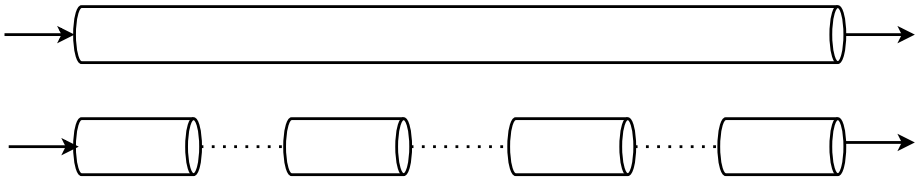


Figure 2.6: Fiber as a concatenation of the small segments of length of Δz.

a partial differential equation, obtained from the general wave equation [2].

Dual-polarization model

Propagation of two signals in the x and y polarizations of the electric field in a single mode fiber is modeled by the coupled nonlinear Schrödinger's equation (CNLSE). Let $q_i(t,z)$ be the complex envelope of the signal in polarization $i \in \{x,y\}$. The CNLSE reads

$$\frac{\partial q_x(t,z)}{\partial z} = -\frac{\alpha}{2}q_x - \beta_{1x}(z)\frac{\partial q_x}{\partial t} - \frac{j\beta_2}{2}\frac{\partial^2 q_x}{\partial t^2} + j\gamma\Big(|q_x|^2 + \frac{2}{3}|q_y|^2\Big)q_x, \tag{2.10}$$

$$\frac{\partial q_y(t,z)}{\partial z} = -\frac{\alpha}{2}q_y - \beta_{1y}(z)\frac{\partial q_y}{\partial t} - \frac{j\beta_2}{2}\frac{\partial^2 q_y}{\partial t^2} + j\gamma\Big(|q_y|^2 + \frac{2}{3}|q_x|^2\Big)q_y, \tag{2.11}$$

where $\beta_{1x}(z)$ and $\beta_{1y}(z)$ are the first-order dispersion coefficients of fiber along the x and y axis of polarizations, respectively. As noted, in birefringent fibers $\beta_{1x}(z) \neq \beta_{1y}(z)$. Thus, the β_{1x} and β_{1y} terms introduce DGD between the two polarization.

In addition, there is also a rotation of the SOP over the surface of the Poincaré sphere along the distance, that is not represented in (2.10)–(2.11). This effect is described separately in the numerical simulation of the propagation equation in Section 2.2.3.

In birefringent fibers where $\mathcal{L} \gg L_B$, the SOP varies rapidly and randomly in distance. In this case, the Manakov-PMD model is obtained by averaging the CNLSE over the SOP

$$\frac{\partial q_i(t,z)}{\partial z} = -\frac{j\beta_2}{2}\frac{\partial^2 q_i}{\partial t^2} + j\frac{8}{9}\gamma\Big(|q_i|^2 + |q_{\bar{i}}|^2\Big)q_i, \tag{2.12}$$

where $\bar{i}$ is the complement of $i \in \{x,y\}$, and we neglected loss and first-order dispersion terms. The Manakov-PMD in the form (2.12) is used in DBP.

2.2.3 Split Step Fourier Method

The SSFM is a numerical method for solving the NLSE. In SSFM, a fiber is viewed as a cascade of segments with length of $\Delta z = \frac{z}{n}$, where $n \to \infty$ is an integer. In each segment, its assumed that the linear and the nonlinear effects act independently. The CNLSE is

then solved for the linear and nonlinear terms separately in each segment, as described below.

Single polarization fiber

The linear part of the single-polarization NLSE (2.9) is

$$\frac{\partial q_l(t,z)}{\partial z} = -\frac{\alpha}{2}q_l - \frac{j\beta_2}{2}\frac{\partial^2 q_l}{\partial t^2}. \tag{2.13}$$

Define $Q_l(\omega, z)$ to be the Fourier transform of the absolutely-integrable function $q_l(t, z)$

$$Q_l(w,z) = \int_{-\infty}^{\infty} q_l(t,z)\exp(j\omega t)dt. \tag{2.14}$$

In the frequency domain (2.13) is

$$\frac{\partial Q_l(w,z)}{\partial z} = -\frac{\alpha}{2}Q_l(w,z) + \frac{j\beta_2 w^2}{2}Q_l(w,z). \tag{2.15}$$

The differential equation (2.15) can be solved in one segment

$$Q_l(\omega,\Delta z) = Q_l(\omega,0)\exp\Big((-\frac{\alpha}{2} + \frac{j\beta_2 w^2}{2})\Delta z\Big), \tag{2.16}$$

where $Q_l(w, 0)$ is the input signal in frequency. The linear step in one segment in SSFM is performed in the frequency domain, by implementing (2.16).

The nonlinear part of the NLSE (2.9) is

$$\frac{\partial q_{nl}(t,z)}{\partial z} = j\gamma|q_{nl}(t,z)|^2 q_{nl}(t,z). \tag{2.17}$$

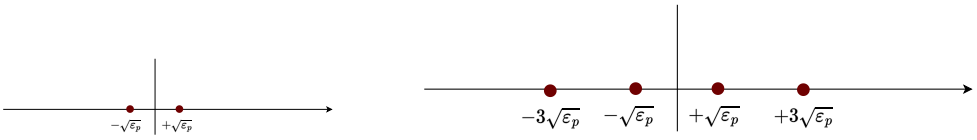
Equation (2.17) can be solved in one segment

$$q_{nl}(t,\Delta z) = q_{nl}(t,0)\exp(j\gamma\Delta z|q_{nl}(t,0)|^2). \tag{2.18}$$

The nonlinear step in one segment in SSFM is performed in the time domain, by implementing (2.18).

Dual polarization fiber

The SSFM can be used to simulate the CNLSE in the dual polarization fiber, with the same procedure described for the single-polarization fiber. However, in the dual-polarization



$M = 2$ $M = 4$

Figure 2.7: PAM constellations with different modulation orders.

case, there is an additional PMD step that is explained below.

Consider a sub-segment of the polarization beat length L_B, *i.e.,* the smallest segment in distance over which the DGD and SOP change. This distance ϵ is typically around one to tens of meters. We assume that the DGD and SOP at the end of this segment are randomly realized, independent of their values at the input of the segment.

Then, the frequency response of the sub-segment i of length L_B can be written as the product of a random unitary matrix for the rotation and a DGD diagonal matrix for the time delay

$$H_i = \begin{bmatrix} \cos\theta_i & \exp\left(-j\frac{\phi_i}{2}\right)\sin\theta_i \\ -\exp\left(j\frac{\phi_i}{2}\right)\sin\theta_i & \cos\theta_i \end{bmatrix} \begin{bmatrix} \exp\left(j\frac{\tau_i}{2}w\right) & 0 \\ 0 & \exp\left(-j\frac{\tau_i}{2}w\right) \end{bmatrix}, \quad (2.19)$$

where τ_i is a sequence of independent identically distributed (iid) random time delays, drawn from a Gaussian distribution with mean zero and variance $\tau_0\sqrt{\epsilon}$, where τ_0 is the PMD parameter. Further, θ_i and ϕ_i are sequences of iid random variables drawn uniformly in $(0, 2\pi]$.

2.3 Digital Optical Modulation

In this section, we briefly review the modulation of digital information in an optical signal.

The input binary data stream is first mapped to a sequence of symbols drawn from a constellation, where each symbol carries multiple bits of information. The symbols are subsequently modulated with a pulse shape. The digital signal is converted into a continuous-time electrical signal, which is then used to drive an optical modulator. The modulated optical signal then propagates in optical fiber.

2.3.1 Digital Modulation

The digital modulation has two stages. In the first stage, a mapper maps a binary stream b_i to a sequence of symbols s_i. The mapping can be memoryless or with memory. In memoryless modulation, the binary sequence is divided into subsequences of a fixed length k, each of which assigned to one symbol in a constellation $\mathcal{C}$ of size $M = 2^k$.

In the Gray mapping, also known as the Gray binary code or the reflected binary code, the Hamming distance between two subsequences assigned to the adjacent symbols is one. This arrangement ensures that if a symbol error occurs during the transmission, most likely a single bit would be flipped. That minimizes the bit error probability during transitions between the adjacent symbols in the constellation.

The second stage is the pulse shaping, where the sequence of symbols is converted to a continuous time baseband waveform. It is assumed that the pulses are transmitted periodically at intervals of T_s seconds. Consequently, during each second, transmissions occur at the baud rate $R_s = \frac{1}{T_s}$ times. The modulated signal is thus

$$q(t) = \sum_{i=\infty}^{\infty} s_i p(t - iT_s),$$

where $p(t)$ is the pulse shape.

In pulse amplitude modulation (PAM), the constellation is a discrete set of real numbers that represent different amplitudes. To ensure a zero mean value in the transmitted signal, the constellation is often chosen to be symmetric around the origin, thus, up to a normalization factor

$$\mathcal{C} := \Big\{\pm 1, \pm 3, \pm 5, \ldots, \pm(M-1)\Big\}.$$

PAM is implemented in the IM/DD (not coherent) systems.

In phase-shift keying (PSK), the input bit stream is mapped to complex numbers with fixed amplitude and distinct phase values, *i.e.,*

$$\mathcal{C} := \Big\{\sqrt{E}\exp\left(j\frac{2\pi}{M}m\right) : \quad m = 0, 1, \cdots, M-1\Big\},$$

where E is the constellation energy.

The quadrature amplitude modulation (QAM) is a two-dimensional modulation format where the points in the constellation can have different real and imaginary values. QAM

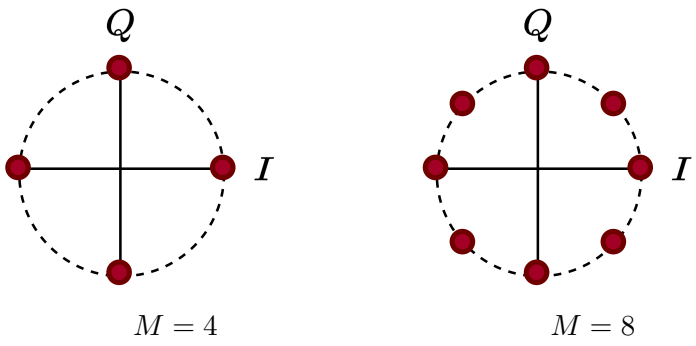


Figure 2.8: phase-shift keying (PSK) constellation with 4 an 8 points.

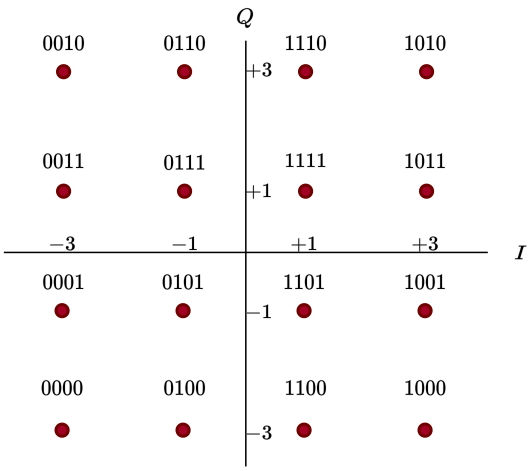


Figure 2.9: 16-QAM constellation with gray coding.

has an in-phase (I) and quadrature (Q) signal, to simultaneously transmit two independent digital streams. The resulting modulated signals are combined to form a single complex-valued signal. QAM is widely used in communications, because it allows for efficient use of bandwidth and high spectral efficiency.

The pulse shape is designed so that the modulated signal aligns well with the characteristics of the transmission medium, and the SNR is maximized. Further, pulse shaping alters the transmitted waveform to meet certain objectives, such as reducing the ISI, improving the spectral efficiency, or minimizing the bandwidth occupied by the signal. Pulse shaping filters are typically applied at both the transmitter and receiver in the communication system. At the receiver, a matched filter based on the pulse shape at the transmitter is used to mitigate the effects of ISI.

The pulse shape can be a sinc function. This is a good filter in theory because it achieves zero ISI and minimizes the distortions. However, it is impractical in real-world applications, because it is discontinuous in the frequency domain and decays slowly in the

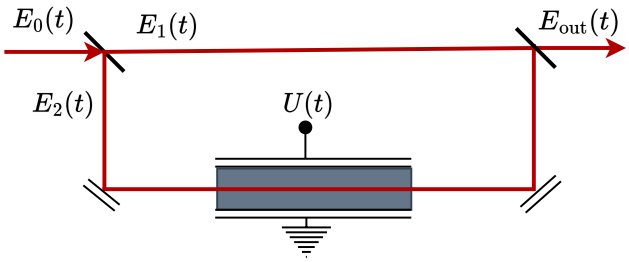


Figure 2.10: Mach Zehnder interferometer.

time domain.

The root raised cosine (RRC) is a practical filter that addresses the limitations of the sinc function. It has an impulse response that decays fast with time, and is continuous in frequency. The impulse response of the RRC filter is

$$h(t) = \begin{cases} \frac{1}{T_s}\left(1+\beta\left(\frac{4}{\pi}-1\right)\right), & t=0 \\ \frac{\beta}{T_s\sqrt{2}}\left[\left(1+\frac{2}{\pi}\right)\sin\left(\frac{\pi}{4\beta}\right)+\left(1-\frac{2}{\pi}\right)\cos\left(\frac{\pi}{4\beta}\right)\right], & t=\pm\frac{T_s}{4\beta} \\ \frac{1}{T_s}\dfrac{\sin\left[\pi\frac{t}{T_s}(1-\beta)\right]+4\beta\frac{t}{T_s}\cos\left[\pi\frac{t}{T_s}(1+\beta)\right]}{\pi\frac{t}{T_s}\left[1-\left(4\beta\frac{t}{T_s}\right)^2\right]}, & \text{otherwise} \end{cases} \tag{2.20}$$

where T_s is the symbol duration, and β is the roll-off factor.

The impulse response of an RRC filter is symmetric around the origin, and real-valued. Thus, the RRC filter serves as a pulse shape as well as its matched filter, without requiring a separate dedicated component.

2.3.2 Optical Modulation

An optical modulator modulates the amplitude of a laser signal, based on an electrical signal. There are two types of them in optical communication: the Mach Zehnder modulator (MZM) and electro-absorption modulator (EAM).

The MZM works by changing the refractive index of an electro-optic material, typically LiNbO3, GaAs or InP, in response to an electrical signal. The MZM has two waveguide arms, with an electro-optic material placed in one arm, as shown in Fig. 2.10 [64]. The light signal is split between the two arms and recombined at the output. By applying a voltage

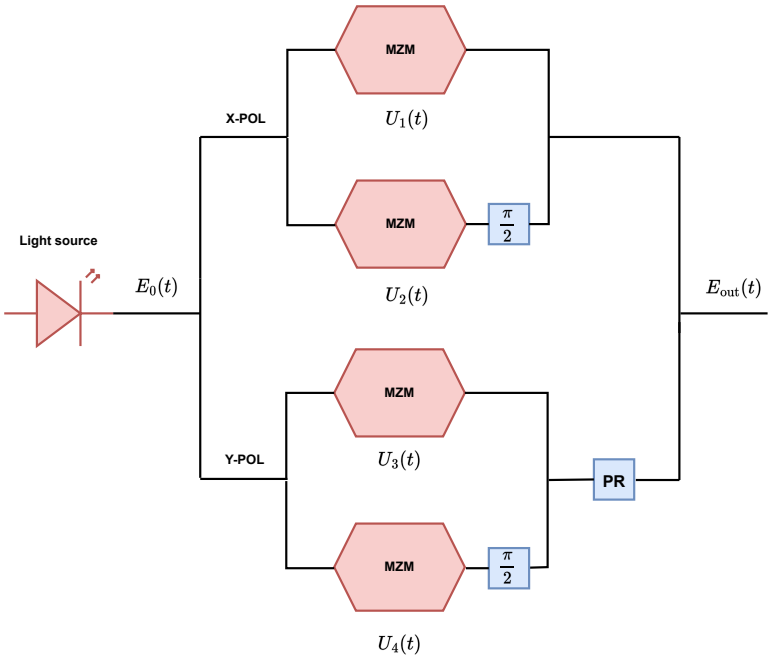


Figure 2.11: MZM for two polarization.

to the material, its refractive index is changed, and the phase of the light passing through that arm is altered. This causes the light to interfere constructively or destructively at the output, thereby modulating the amplitude of the optical signal.

The complex envelope of the optical signal at the output of the modulator is

$$E_{out}(t) = E_1(t)\exp(-j\theta_1(t)) + E_2(t)\exp(-j\theta_2(t)), \tag{2.21}$$

where $\theta_1(t)$ (resp. $E_1(t)$) and $\theta_2(t)$ (resp. $E_2(t)$ are the phase shifts (resp. amplitudes) in the first and the second arm of the modulator. The output power is

$$P_{out} = P_1 + P_2 + 2\sqrt{P_1 P_2}\cos\left(\theta_1(t) - \theta_2(t)\right), \tag{2.22}$$

where P_1 is the power of the signal traveling in the first arm, and P_2 is the power of the signal going thought the material.

A single MZM with a two-level electrical signal is used to produce an on-off keying intensity modulated optical signal [66]. A dual-drive MZM comprises two single MZM, as shown in Fig. 2.11. This configuration enables modulation of the real and imaginary components. This allows modulation in-phase and quadrature, for example, via QPSK or QAM. The diagram of a dual-parallel MZM is depicted in Fig. 2.11.

The EAM, on the other hand, works based on the Franz–Keldysh effect [130]. It relies

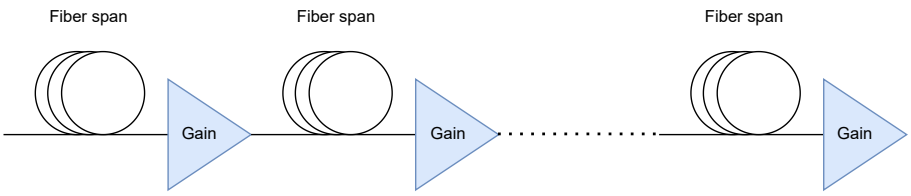


Figure 2.12: A fiber transmission link with EDFA.

on the changes in the absorption level of a material in response to an electrical signal. The material absorbs the light passing through it based on the amplitude of an applied electrical voltage, resulting in optical modulation. However, the chirp-induced modulation introduces a phase distortion. Thus, EAM is not suitable for coherent transmission.

2.4 Optical Amplification

Optical amplification compensates for the attenuation in the fiber, without the need for excessive regeneration along the distance, or optical to the electrical conversion. In this Section, we review discrete amplification with EDFA and continuous Raman amplification.

2.4.1 Erbium Doped Fiber Amplifier

In discrete amplification, the transmission link is divided into a number of fiber spans separated by EDFAs, as shown in Fig. 2.12. The length of each span varies depending on the system configuration and the type of fiber, but typically is between 50 and 100 km in terrestrial systems.

The EDFA [9] operates based on the principle of the stimulated emission. A piece of optical fiber is doped with the erbium. As the light travels through the doped fiber, it stimulates the erbium atoms, which then emit photons at the same wavelength as the incoming photons. This process amplifies the optical signal.

The EDFA has a pump, which provides the necessary energy for amplification. A wavelength-division multiplexer combines the optical pump signal with the incoming light signal in the erbium-doped fiber. An isolator ensures unidirectional transmission and prevents the reflection of light that could introduce noise and inefficiency. These components must meet certain requirements such as low insertion loss, polarization insensitivity, and high stability to ensure good performance.

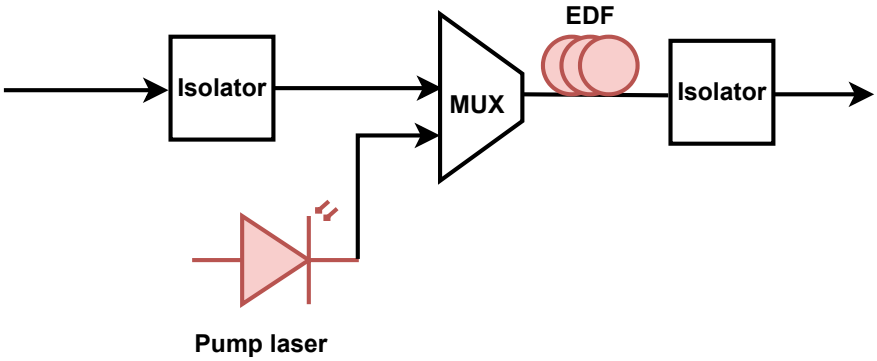


Figure 2.13: The structure of the EDFA.

2.4.2 Raman Amplifier

There are two types of Raman amplifiers: forward- and backward-pumped. In a forward-pumped Raman amplifier, a high-power pump laser is coupled into the fiber-optic cable along with the signal. The pump laser creates stimulated Raman scattering, which amplifies the signal as it travels through the fiber. In a backward-pumped Raman amplifier, the pump laser is located at the end of the fiber, and the signal travels through the fiber in the opposite direction. The pump laser creates a counter-propagating wave that amplifies the signal.

Raman amplifiers have several advantages over EDFA. They have a wider gain bandwidth, lower noise figure, and have a polarization-independent gain.

2.4.3 Amplification Noise

There are several types of noise in communication systems. An optically amplified signal is subject to the ASE. Thermal or electronics noise arises from the random motion of the electrons in circuits. Shot noise, which is produced by light sources, pertains to the randomness in the arrival time of the photons. However, in optical transmission, the ASE is the dominant source of the noise.

The ASE noise $n(t)$ is assumed to be a band-limited white circularly symmetric Gaussian stochastic process, *i.e.,* with the autocorrelation function

$$E\{n(t)n^*(t')\} = \sigma^2\delta_B(t-t'),$$

where $\delta_B(x) = B\text{sinc}(Bx)$, and σ^2 is the noise power spectral density (PSD). For EDFA,

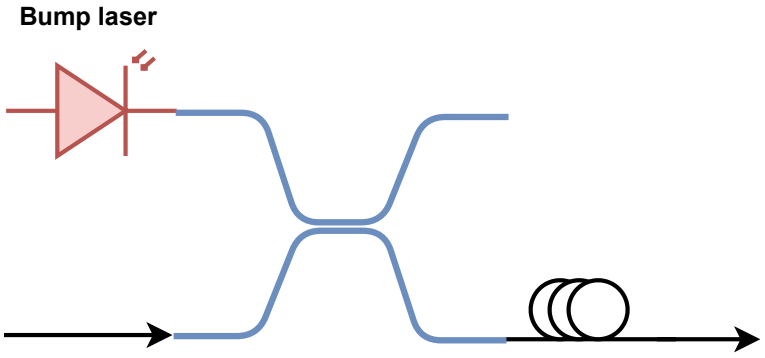


Figure 2.14: The structure of the Raman amplifier.

the noise PSD is

$$\sigma^2 = \frac{1}{2}(G-1)hf_0\mathrm{NF}, \tag{2.23}$$

where h is the Planck's constant, f_0 is the center frequency, and NF is the noise figure. The amplifier gain G for a span of length L_{sp} is

$$G = \exp(\alpha L_{sp}). \tag{2.24}$$

The noise $n(t,z)$ introduced by Raman amplification is similarly defined. The noise is introduced continuously in distance, and has the autocorrelation function

$$E\{n(t,z)n^*(t',z')\} = \sigma^2\delta_B(t-t')\delta(z-z'),$$

where δ is Dirac Delta function. The noise PSD is

$$\sigma^2 = \alpha h f_0 n_{sp}, \tag{2.25}$$

where n_{sp} is the spontaneous emission factor.

2.5 Digital Coherent Receiver

The optical signal is converted to the electrical signals using the coherent detection. The coherent receiver has a 90-degree hybrid mixer for each polarization that mixes the optical signal with a local oscillator, followed by a balanced photo-diode system, to produce the intermediate-frequency in-phase and quadrature components of the signals, as shown in Fig. 2.15. The electrical signals are amplified and digitized by the analog-to-digital con-

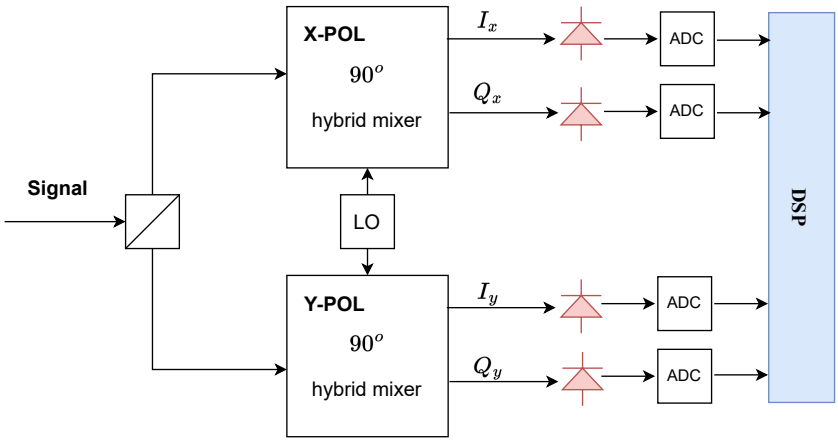


Figure 2.15: Coherent receiver in two polarizations.

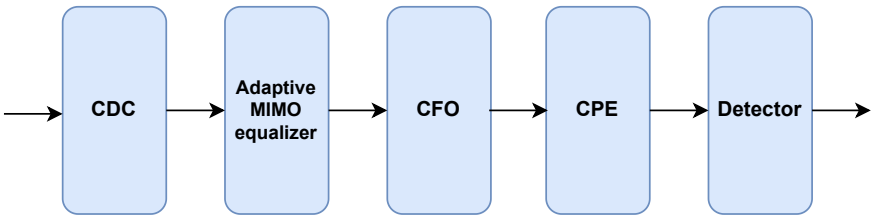


Figure 2.16: DSP in the digital coherent receiver.

verters. The digitized signals are then processed using DSP, to compensate for the fiber impairments and recover the transmitted data. Since the optical signal is converted into the electrical domain, the phase and polarization tracking are also performed in the digital domain [100].

The schematic diagram of DSP in the coherent receiver is shown in Fig. 2.16. In the following, we describe each block in this diagram.

2.5.1 Chromatic Dispersion Compensation

The chromatic dispersion is static, and independent of the polarization. It can thus be compensated at the beginning of the DSP chain at the RX, before the signal is demultiplexed into two orthogonal polarization states.

The frequency response of the CD filter is obtained by changing the sign of the β_2 term

$$H_{CD}(w) = \exp\left(-\frac{j\beta_2 w^2 z}{2}\right). \tag{2.26}$$

This is a zero forcing equalizer, which is optimal in this case since the CD does not lead to noise enhancement.

Alternatively, the equalization can also be done in the time domain. The impulse

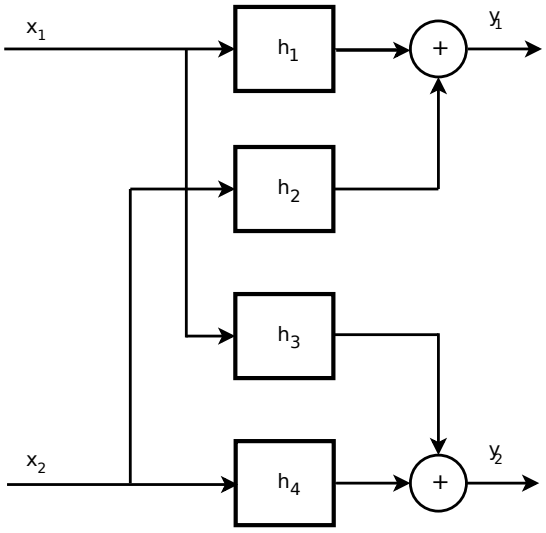


Figure 2.17: Schematic diagram of the multiple input multiple output (MIMO) equalizer.

response of the filter can be obtained by inverting (2.26)

$$h_{CD}(t) = \sqrt{\frac{-j}{2\pi\beta_2 z}} \exp\left(\frac{-j}{2\beta_2 z} t^2\right). \tag{2.27}$$

Different digital filters have been used to compensate the effects of CD in the time and frequency [99]. The frequency domain equalizers have become increasingly favored over the time domain ones based on the finite frequency response (FIR) or adaptive least mean square filters. This is due to their low computational complexity, especially for large accumulated dispersion, and their suitability for varying fiber distances [122].

2.5.2 Adaptive MIMO Equalizer

The polarization dependent effects are time varying. Therefore, their compensation is adaptive. The transfer function of the multiple input multiple output (MIMO) equalizer is a 2×2 frequency-selective matrix. Denote the inputs of the MIMO equalizer by $\mathbf{x}_1[k]$ and $\mathbf{x}_2[k]$, corresponding to the signals of the x and y polarization at the integer time steps k. The corresponding outputs $\mathbf{y}_1[k]$ and $\mathbf{y}_2[k]$ are

$$\begin{bmatrix} \mathbf{y}_1[k] \\ \mathbf{y}_2[k] \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1[k] & \mathbf{h}_2[k] \\ \mathbf{h}_3[k] & \mathbf{h}_4[k] \end{bmatrix} \begin{bmatrix} \mathbf{x}_1[k] \\ \mathbf{x}_2[k] \end{bmatrix}, \tag{2.28}$$

where $\mathbf{h}_1$,$\mathbf{h}_2$,$\mathbf{h}_3$ and $\mathbf{h}_4$ are FIR filters of length N.

The MIMO equalizer filter taps are estimated by the constant modulus (CM) (for the PSK) or the radially directed equalizer (RDE) (for 16-QAM). The update equations for

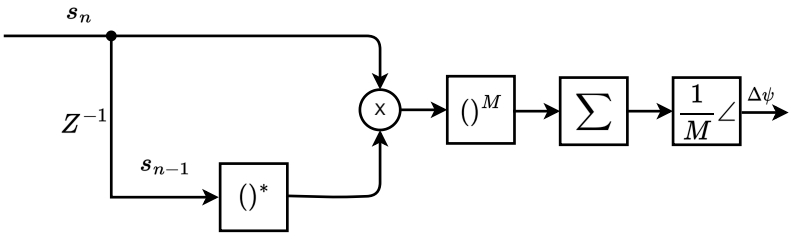


Figure 2.18: Schematic diagram of the frequency estimator.

the filter taps are

$$\mathbf{h}_1[k+1] = \mathbf{h}_1[k] - \mu\Big(|\mathbf{y}_1[k]|^2 - R_1\Big)\mathbf{y}_1[k]\mathbf{x}_1[k], \tag{2.29}$$

$$\mathbf{h}_2[k+1] = \mathbf{h}_2[k] - \mu\Big(|\mathbf{y}_1[k]|^2 - R_1\Big)\mathbf{y}_1[k]\mathbf{x}_2[k], \tag{2.30}$$

$$\mathbf{h}_3[k+1] = \mathbf{h}_3[k] - \mu\Big(|\mathbf{y}_2[k]|^2 - R_2\Big)\mathbf{y}_2[k]\mathbf{x}_1[k], \tag{2.31}$$

$$\mathbf{h}_4[k+1] = \mathbf{h}_4[k] - \mu\Big(|\mathbf{y}_2[k]|^2 - R_2\Big)\mathbf{y}_2[k]\mathbf{x}_2[k], \tag{2.32}$$

where μ is the learning rate, and for a normalized input

$$R_i = \begin{cases} 0.2, & |\mathbf{y}_i[k]| < \frac{1+\sqrt{0.2}}{2}, \\ 1.8, & |\mathbf{y}_i[k]| > \frac{1+\sqrt{1.8}}{2}, \\ 1, & \text{else}, \end{cases} \tag{2.33}$$

for $i \in \{1, 2\}$.

We point out that the FIR filters in the MIMO equalizer can compensate the residual CD as well. This implies that in optical networks that employ dispersion management through periodically placed dispersion compensating modules, a simpler DSP without a dedicated CD compensation unit would suffice [101].

2.5.3 Carrier Frequency Estimation

In communication systems, the transmitted signal may experience frequency offsets due to a variety of factors caused by the propagation environment. In the optical transmission, the frequency mismatch Δf between the lasers at the transmitter and receiver is called the carrier frequency offset (CFO).

There are various algorithms in digital communications for frequency offset estimation. We consider a blind feed-forward differential phase estimation algorithm that does not rely on data, illustrated in Fig. 2.18. This algorithm is particularly suited to implementation

in a high speed signal processor [71].

The objective of the phase/frequency estimator is to determine the phase difference $\Delta\psi$ between two successive samples s_n, s_{n-1}

$$\Delta\psi = 2\pi\Delta f T_0, \tag{2.34}$$

where T_0 is the sampling time. The correction process is conducted in the following manner. Initially, the received symbol is multiplied by the complex conjugate of the previous symbol, resulting in a complex number whose phase equals to the phase difference between the two symbols. Next, the information in the signal phase is eliminated. For the PSK signals, this can be accomplished by raising the complex symbol to the power of the number of constellation points. The outcome is averaged over a large number of samples. The phase is then divided by the modulation order, resulting in an estimate of the phase difference between the consecutive symbols. To correct for the frequency offset, a running symbol index is used to subtract the accumulated phase offset from each symbol, obtaining a corrected symbol

$$\psi_n = n\Delta\psi T_0. \tag{2.35}$$

2.5.4 Constant Phase Estimation

The constant phase estimation (CPE) is carried out at the end, after the compensation of the channel impairments. The algorithm will simultaneously test different carrier phase angles and determine the most likely one among these.

The phase values are

$$\xi_b = \frac{\pi}{2}\Big(\frac{b}{B} - \frac{1}{2}\Big), \tag{2.36}$$

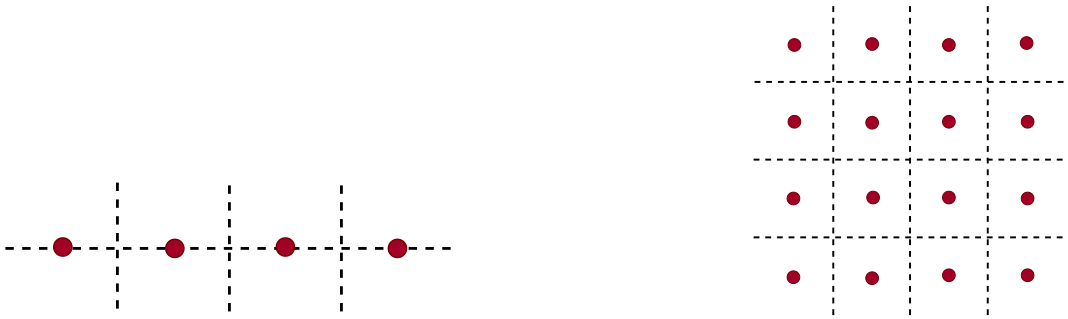
where $b \in \big\{0, 1, 2, \cdots, B-1\big\}$. The phase-rotated symbols y_i are fed to a decision circuit. First, the metric D_b is calculated

$$D_b = \sum_{i=-N}^{N} \left[y_i \exp\left(-j\xi_b\right) - \lfloor y_i \exp\left(-j\xi_b\right)\rceil\right], \tag{2.37}$$

where $2N+1$ is the number of symbols. The decision rule is

$$\hat{b} = \arg\min_b(D_b). \tag{2.38}$$

The phase $\xi_{\hat{b}}$ corresponding to $\hat{b}$ that minimizes D_b is chosen.



(a) 4 PAM (b) 16 QAM

Figure 2.19: The decision regions of PAM for the AWGN channel and the ML rule.

2.5.5 Detection

Once the received soft symbols are demodulated in a vector $\mathbf{r} = (r_1, r_2, \cdots, r_M)$, the detection unit in the receiver makes a decision on which symbol was transmitted. The decision function used by the receiver is denoted as $D(\mathbf{r})$, which is a function mapping $\mathbf{r}$ to the set of transmitted symbols $(s_1, s_2, \cdots, s_M)$.

The probability of the decision $D(\mathbf{r}) = s_{\hat{m}}$ is correct is the probability that $s_{\hat{m}}$ was indeed the transmitted message. The objective is to find an optimal detector that minimizes the error probability or equivalently, maximizes the probability of a correct decision

$$\begin{aligned} \hat{s}_m &= \arg \max_{1 \le m \le M} P[s_m|\mathbf{r}] \\ &:= D^*(\mathbf{r}), \end{aligned}$$

where $P[s_m|\mathbf{r}]$ is the conditional probability distribution of s_m given $\mathbf{r}$. The decision rule given by the above equation is the maximum a posteriori probability (MAP) rule.

The MAP rule can be rewritten as,

$$\hat{s}_m = \arg \max_{1 \le m \le M} \frac{P_{s_m} P[\mathbf{r}|s_m]}{P_{\mathbf{r}}}, \tag{2.39}$$

where P_{s_m} and P_r are the probability of the transmitted symbol s_m and the demodulated symbols $\mathbf{r}$, respectively. If the messages are assumed to be equally probable, the MAP rule is reduced to:

$$\hat{s}_m = \arg \max_{1 \le m \le M} \frac{P[\mathbf{r}|s_m]}{P_{\mathbf{r}}} \tag{2.40}$$

$$= \arg \max_{1 \le m \le M} P[\mathbf{r}|s_m], \tag{2.41}$$

where the last equality follows since $P_{\mathbf{r}}$ is constant with respect to m. The above detector is called the maximum likelihood (ML) rule. The ML receiver is not optimal unless the transmitted symbols have equal probabilities. Nonetheless, it remains a popular choice, since obtaining information about the message probabilities can be difficult.

The detector divides the output space $\mathbb{C}^M$ into M regions $R_1, R_2, ..., R_M$, where R_m is the decision region corresponding to the message s_m. If $\mathbf{r} \in R_m$, then the detector decides in favor of the message s_m and outputs $\hat{s}_m = D(\mathbf{r})$. For the ML rule, the decision regions are

$$R_m = \Big\{ \mathbf{r} \in \mathbb{C}^M : P[\mathbf{r}|s_m] > P[\mathbf{r}|\tilde{s}_m], \quad \forall \tilde{s}_m \in (s_1, s_2, \cdots, s_M), \;\; s_m \neq \tilde{s}_m \Big\}. \tag{2.42}$$

An error occurs when the output $\mathbf{r}$ is not in R_m giving that s_m was transmitted. Hence, the probability of the symbol error can be stated as:

$$P_e = \sum_{m=1}^{M} P_{s_m} P[\mathbf{r} \notin R_m | s_m \text{ was sent}]. \tag{2.43}$$

Equation (2.43) can be also be written as

$$P_e = \sum_{m=1}^{M} P_{s_m} \sum_{\hat{m}=1, \hat{m} \neq m}^{M} \int_{R_{\hat{m}}} P[\mathbf{r}|s_m] d\mathbf{r}. \tag{2.44}$$

This is the expression for the symbol error probability, *i.e.,* the likelihood of an error occurring during the transmission of a symbol.

The error may also be measured in terms of the bit error probability P_b. In general, calculating the bit error probability requires an understanding of how various bit sequences are mapped to symbols. Hence, determining the bit error probability can be more difficult. The derivation is however simplified if the constellation displays certain symmetry properties. Moreover, we can bound the bit error probability by observing that a symbol error occurs when at least one bit is erroneous, thus, $P_b \leq P_e$. Further, the event of a symbol error is the union of the events of the errors in the k bits that represent that symbol. Therefore, from the union bound, $P_e \leq kP_b$. Combining these bounds, $P_b \leq P_e \leq KP_b$.

2.6 Nonlinearity Mitigation

2.6.1 Digital Backpropagation

The digital back-propagation is used to compensate the linear (mainly CD) and nonlinear effects in fiber using DSP. The dual-polarization DBP uses the SSFM to solve the inverse of the Manakov-PMD equation (2.12), approximating the transmitted signal from the received one. We assume that the DBP has the knowledge of the CD and nonlinearity parameters β_2 and γ.

The fiber link is partitioned into n segments of length Δz. The linear operator of DBP compensating the CD in a small segment in the frequency domain is

$$L_{\text{DBP}} = \exp\left(-\frac{j\beta_2 w^2 \Delta z}{2}\right) I_2. \tag{2.45}$$

where I_2 is 2×2 identity matrix. The nonlinear operator is

$$N_{\text{DBP}} = \exp\left(-j\frac{8}{9}\gamma\Big(|q_x(t,.)|^2 + |q_y(t,.)|^2\Big)\Delta z\right) I_2. \tag{2.46}$$

where $q_x(t,.)$ and $q_x(t,.)$ are the signals of the x and y polarization in the time domain at the input of the segment.

2.6.2 Volterra Based Equalizer

The Volterra series provides a generalization of the impulse response representation of the linear systems to the nonlinear systems. For transmission in one polarization, the Volterra series expresses the output signal $q(t,z)$ at distance z in terms of the input signal $q(t,0)$ as follows [129]:

$$\begin{aligned} q(t,) = &\int_{-\infty}^{\infty} h^{(1)}(\tau) q(t-\tau,0) d\tau \\ &+ \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h^{(3)}(\tau_1,\tau_2,\tau_3) q(t-\tau_1,0) q^*(t-\tau_2,0) q(t-\tau_3,0) d\tau_1 d\tau_2 d\tau_3, \end{aligned} \tag{2.47}$$

where $h^n(.)$ is the nth order Volterra kernel, * is the conjugate, and we ignored the higher-order terms beyond cubic.

The Volterra series is often implemented in the frequency domain for nonlinearity com-

pensation in the optical fiber communication. In the frequency domain, (2.47) is

$$Q(\omega, z) = h^{(1)}(\omega)Q(\omega, 0) + \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} h^{(3)}(\omega, \omega_1, \omega_2, \omega_3)Q(\omega_1, 0)Q^*(\omega_2, 0)Q(\omega_3, 0)\delta_{123}(\omega, \omega_1, \omega_2, \omega_3)d\omega_1 d\omega_2 d\omega_3, \tag{2.48}$$

where

$$\delta_{123w} := \delta(w_1 - w_2 + w_3 + w). \tag{2.49}$$

where $\delta(.)$ is Dirac Delta function.

For the normalized NLSE with $\beta_2 = -2$ and $\gamma = 2$, the first and the third-order Volterra kernels are given by

$$h^{(1)}(w) = e^{jw^2 z}, \tag{2.50}$$

and

$$h^{(3)}(w, w_1, w_2, w_3) = -jze^{-jz\frac{1}{2}(w_2 - w_1 + w_2 - w_3)}\text{sinc}\left(\frac{1}{2}z(w_2 - w_1 + w_2 - w_3)\right), \tag{2.51}$$

where $\text{sinc}(x) = \frac{\sin(x)}{x}$.

The Volterra equalizer inverts the fiber channel by inverting the Volterra kernels. The performance of DBP and Volterra equalizer are compared in [77, 6]. In some cases, DBP requires more FFT operations compared to the Volterra equalizer, and is thus more computationally complex.

In the next chapter, we will introduce neural networks (NNs), that will be used later to compensate the fiber nonlinearities.

CHAPTER 3

Introduction to Neural Networks

This chapter provides a brief introduction to the main concepts in NNs, relevant to this dissertation. We will begin by recalling the empirical risk minimization framework, that formalizes the learning by the NNs. We will introduce the architectures used in the subsequent chapters of this thesis, namely, the MLP, convolutional and recurrent models. Further, we will review the training of the NNs using the stochastic gradient descent (SGD), as well as the backpropagation algorithm for the calculation of the gradient of the loss function with respect to the weights and biases. Lastly, we will mention a few applications where the NNs have achieved the state-of-the-art results.

3.1 Statistical Learning Framework

The statistical machine learning provides a theoretical framework for learning from data. The assumption is that data is generated by some underlying process, and that this process can be modeled using statistical models. The goal of the statistical learning is to obtain a model that makes good predictions on unseen data.

The learning algorithm predicts the label of an input object in a given domain set. It is assumed that the learner has access to three components.

- *Domain set* $\mathcal{X}$: This is the set of objects that the learner wishes to label. Typically, the domain points are represented by vectors of features that describe the object.

- *Label set* $\mathcal{Y}$: This is the set of possible labels that the learner can assign to each point in the domain set.

- *Training data*: This refers to a finite sequence of input output pairs

$$S = \left\{(\mathbf{x}^{(1)}, \mathbf{y}^{(1)}), \cdots, (\mathbf{x}^{(n)}, \mathbf{y}^{(n)})\right\}, \quad \mathbf{x}^{(i)} \in \mathcal{X}, \quad \mathbf{y}^{(i)} \in \mathcal{Y}.$$

  The training data thus consists of the labeled domain points.

The learner aims to produce a prediction rule or hypothesis $h : \mathcal{X} \to \mathcal{Y}$. This predictor can be utilized to forecast the label of the unseen points in $\mathcal{X}$.

It is assumed that the input points are generated from a probability distribution $\mathcal{D}$ over $\mathcal{X}$. It is essential to note that we do not make any assumptions regarding the learner's knowledge of this distribution. Regarding the labels, it is assumed that there exists a "correct" labeling function $f : \mathcal{X} \to \mathcal{Y}$, where $\mathbf{y}^{(i)} = f(\mathbf{x}^{(i)})$ for all i. However, the learner is not aware of this labeling function; in fact, figuring out this function is precisely what the learner is attempting to accomplish.

The error of a prediction rule $h : \mathcal{X} \to \mathcal{Y}$ is defined as the likelihood that it will not correctly predict the label of a random data point generated from the underlying distribution. In other words, the error of h is the probability of selecting a random instance $\mathbf{x}^{(i)}$, drawn from the distribution $\mathcal{D}$, where $h(\mathbf{x}^{(i)})$ does not match $f(\mathbf{x}^{(i)})$

$$L_{\mathcal{D},f}(h) := \Pr_{\mathbf{x}^{(i)} \sim \mathcal{D}}[h(\mathbf{x}^{(i)}) \neq f(\mathbf{x}^{(i)})] := \mathcal{D}(\mathbf{x}^{(i)} : h(\mathbf{x}^{(i)}) \neq f(\mathbf{x}^{(i)})). \tag{3.1}$$

The error of the predictor $L_{\mathcal{D},f}$ is also known as the generalization error.

The learner lacks the knowledge of the underlying distribution $\mathcal{D}$, and the labeling function f. The only means for the learner to engage with the environment is by examining a training data set. Therefore, a suitable strategy is to search for a predictor that performs well on the training data. This approach is commonly referred to as the empirical risk minimization (ERM). The error in this scenario is defined as

$$L_S(h) := \frac{1}{n} \left| \left\{ k : \ h(\mathbf{x}^{(k)}) \neq \mathbf{y}^{(k)} \right\} \right|. \tag{3.2}$$

The hypothesis that minimizes this error in a class is called the ERM rule [106].

While the ERM rule may seem intuitive, it has the potential to fail drastically if not implemented carefully. Overfitting is a phenomenon where a predictor performs exceptionally well on the training set, but performs poorly on the unseen data. Essentially, overfitting occurs when the hypothesis fits the training data too closely. To mitigate the

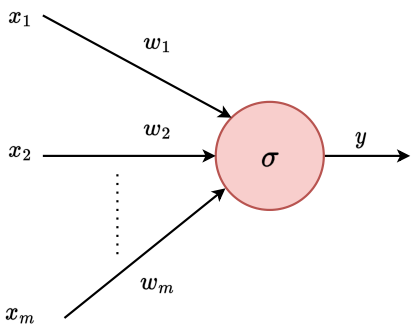


Figure 3.1: The diagram of a neuron.

problem of overfitting, one approach is to limit the search space of the ERM rule. This could be achieved by pre-selecting a set of predictors or a hypothesis class $\mathcal{H}$. The ERM rule is thus given by

$$h^* = \arg\min_{h \in \mathcal{H}} L_S(h), \tag{3.3}$$

where it is assumed that the minimum achievable.

The bias-variance (or bias-complexity) trade-off refers to the relationship between the learning algorithm's bias and complexity. Bias refers to the algorithm's ability to learn and model arbitrary patterns in the data, while complexity is related to the model's ability to fit the training data. The trade-off arises because increasing the model complexity tends to reduce the bias causing overfitting, while decreasing the complexity increases bias leading to underfitting. The quantitative definitions of bias and variance can be found in [106].

When choosing a hypothesis class $\mathcal{H}$, it is necessary to find a balance between the bias and complexity that allows the algorithm to best generalize to new data. Typically, this balance is achieved by ensuring that the number of the training examples $|S|$ is much greater than the size of the hypothesis class $|\mathcal{H}|$. However, it has been observed that deep learning violates the classical bias-complexity trade-off, and still achieves excellent generalization performance [131, 91].

3.2 Neural Networks

Before providing a definition for a NN, first we define a neuron which serves as the building block for NNs [47]. A neuron takes several inputs $x_i \in \mathbb{R}$ and produces a single output y, according to the following equation

$$y = \sigma\Big(\sum_{i=1}^{m} w_i x_i + b\Big), \tag{3.4}$$

where $w_i \in \mathbb{R}$, $i = 1, 2, \cdots, m$, are the weights, $b \in \mathbb{R}$ is the bias, and σ is the activation function.

A NN is an interconnected group of neurons, aggregated into layers. Different layers may perform different transformations on their inputs. The input travels from the first layer (the input layer) to the last layer (the output layer). The input layer actually does not perform any function on the input; it merely forwards the input to the next layer. The layers between the input and output layer are called the hidden layers and perform transformations on their inputs.

By setting the weights, biases and the activation function, a certain output will be generated. The NN learns the mapping between the input and output by adjusting its weights and biases.

3.2.1 Activation Functions

The activation function takes the weighted sum of the inputs to a neuron and applies a nonlinear transformation to produce the output of the neuron. The purpose of the activation is to introduce nonlinearity in the output, allowing the NN to model complex nonlinear relationships between the input and output. Without activation functions, a NN would simply be a linear regression model, which cannot model complex patterns in the data.

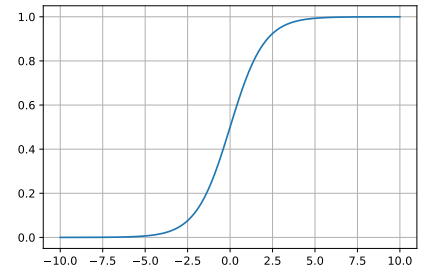
There are several activation functions commonly used in deep learning. Examples used in this dissertation are the sigmoid, rectified linear unit (ReLU), and hyperbolic tangent (Tanh). These activations are shown in Fig. 3.2.

The choice of the activation function depends on the problem and the architecture of the NN. ReLU is widely used, because it is simple to compute and has a gradient bounded away from zero for positive input. On the other hand, sigmoid takes values in $[0, 1]$ and can naturally represent the probability of a class. Both sigmoid and Tanh are used in the recurrent networks. In equalization in fiber-optic communication, Tanh often works better.

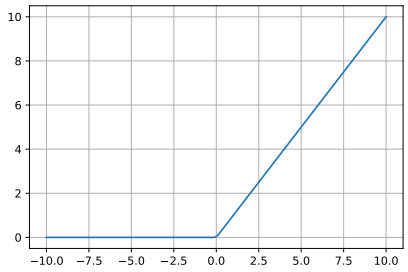
3.2.2 Architectures

Multi-layer perceptron

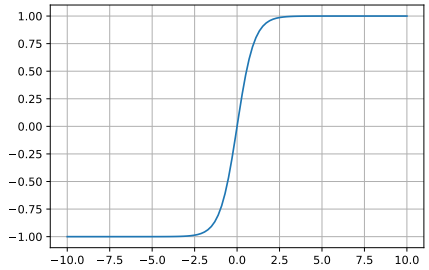
A dense layer is one that is fully connected to the previous layer. Each neuron in a dense layer is connected to every neuron in the previous layer. This type of layer is widely used in neural networks. The MLP (or a dense NN) is a NN where all layers are dense.



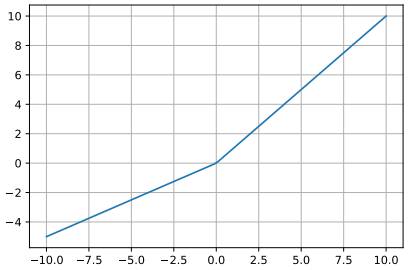
(a) Sigmoid



(b) ReLU



(c) Tanh



(d) Leaky ReLU

Figure 3.2: Several activation functions commonly used in NNs.

Suppose that a dense layer with m_1 neurons follows any layer with m_2 neurons. If $\mathbf{x} = [x_1, x_2, \cdots, x_{m_2}]^T$ is the output of the first layer, and $\mathbf{y} = [y_1, y_2, \cdots, y_{m_1}]^T$ the output of the second layer, then

$$\begin{aligned} \mathbf{y} &= \begin{bmatrix} w_{1,1}x_1 + w_{1,2}x_2 + \cdots + w_{1,m_2}x_{m_2} \\ w_{2,1}x_1 + w_{2,2}x_2 + \cdots + w_{2,m_2}x_{m_2} \\ \vdots \\ w_{m_1,1}x_1 + w_{m_1,2}x_2 + \cdots + w_{m_1,m_2}x_{m_2} \end{bmatrix} \\ &= W\mathbf{x}, \end{aligned} \tag{3.5}$$

where

$$W = \begin{bmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,m_2} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,m_2} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m_1,1} & w_{m_1,2} & \cdots & w_{m_1,m_2} \end{bmatrix}.$$

The weight matrix $W \in \mathbb{R}^{m_1 \times m_2}$ is dense, *i.e.*, in general, w_{ij} are arbitrary non-zero values for all i, j.

An intuitive explanation of (3.5) is as follows. The neurons in a dense layer take an input from every neuron in the previous layer and perform matrix-vector multiplication. This multiplication involves matching the row vector of the output from the previous layer with the column vector of the dense layer, following the rule that the row vector must have the same number of columns as the column vector.

The dense layers are powerful, however, often give rise to overfitting. This motivates lower complexity layers discussed next.

Convolutional Neural Networks

A convolutional layer performs the mathematical operation of the convolution between its input and a kernel or filter. The convolution is obtained by sliding the filter over an input of the same shape, and each time computing the dot product between the two (namely, sum of the element-wise multiplication of the filter taps and the corresponding input values at each position). The output of the convolution is a feature map, which contains information about the presence of a specific feature or pattern at different locations in the input data.

In one dimension, the convolution of an input sequence $\mathbf{x} = (x_1, \cdots, x_m)$ of length m with a kernel or filter $\mathbf{h} = (h_1, \cdots, h_k)$ of length $k \leq m$ is

$$(\boldsymbol{x} * \mathbf{h})(\ell) := \sum_{i=1}^{k} x_{i+\ell-1} h_i, \quad \ell = 1, \cdots, m-k+1, \tag{3.6}$$

where the symbol * denotes the convolution operation.

The convolution in dimensions bigger than one can be defined using the following visualization. The output of the convolution operation at each position ℓ can obtained as follows: shift the filter $\mathbf{h}$ by ℓ positions, multiply that element-wise with the input $\mathbf{x}$, and sum the resulting values. This operation is continued until the filter scans the entire input.

The convolution in (3.6) has stride 1, and the so-called "valid padding". The stride is the step size of the shift of the filter over the input. The relation (3.6) can be straightforwardly extended to the case with stride $s > 1$, and different paddings that determine the size of the output. In this thesis, we frequently use the "same padding," where the input is padded with zeros on the boundaries such that the output has the same size as the input [47].

A convolutional layer typically has more than one filter, each of which learns to detect a particular feature or pattern in the input data. The use of the convolutional layers in the multi-layer NNs allows them to learn hierarchical representations of the features in data.

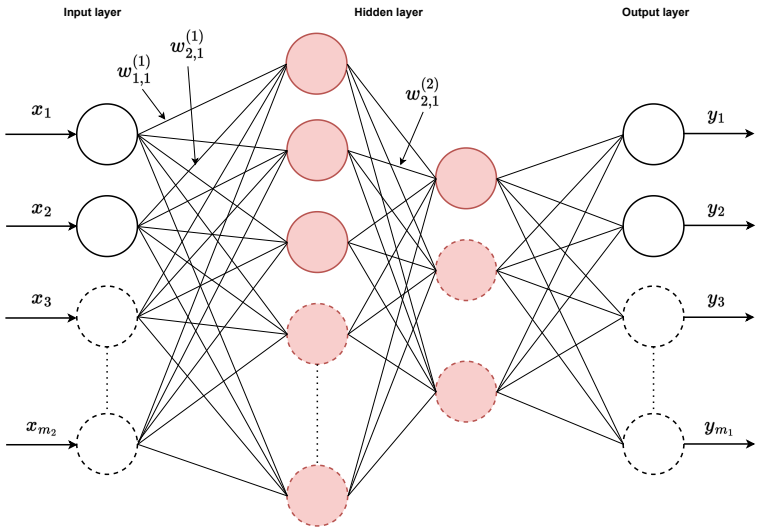


Figure 3.3: Multi-layer perceptron.

For example, in image classification, the initial layers detect simple features like the edges and corners, while the subsequent layers detect more complex patterns like the shapes and objects.

During the training process, the network learns to adjust the values of the filters so as to minimize the difference between the predicted output and the true output. The training of the convolutional NNs is application-specific, to some extent, and beyond the scope of this thesis. The reader is referred to [47] for this.

Recurrent and Long Short Term Memory Networks

The recurrent neural networks are a class of NNs where the connections between the neurons form cycles. The output of a neuron at a given "time step" t feeds the input of that neuron in the next step $t+1$. The recurrent neural network (RNN) uses an internal state to store the long-term temporal information. The network exhibits temporal dynamics, suitable for prediction with sequential data such as the time series.

The operation of a RNN in one step (see (3.7)) can be unfolded from $t = 1$ to $t = T$ in a computational graph similar to that of a multi-layer NN. As T is increased, the hidden state cycles around the network's recurrent connections, and the network's output may vanish or blow up. The RNN is thus prone to the vanishing or exploding gradient problem.

An LSTM is a recurrent network designed to address the problem of the vanishing gradient in the training. An LSTM unit (or cell) at the time step t has an input $\mathbf{x}^{(t)} \in \mathbb{R}^{m_x}$, a hidden state $\mathbf{h}^{(t)} \in (-1,1)^{m_h}$, a cell state $\mathbf{c}^{(t)} \in (-1,1)^{m_h}$, and a cell activation state

$\tilde{\mathbf{c}}^{(t)} \in (-1,1)^{m_h}$. The unit incorporates three gates to process the memory, each of which a dense layer with the sigmoid activation. There is an input (or update) gate, an output gate and a forget gate, with activations $\mathbf{\Gamma_i}^{(t)}, \mathbf{\Gamma_o}^{(t)}, \mathbf{\Gamma_f}^{(t)} \in (0,1)^{m_h}$, respectively.

The forget gate determines the information that is retained from the previous to the next cell state. The closer the output of this gate is to 1, the more the information is retained. Similarly, the input gate determines the information that is retained from the cell activation state to the cell state. Lastly, the output gate learns the values in the current cell state that should be kept as the output of the hidden state. The diagram of an LSTM unit is shown in Fig. 3.4.

The LSTM has a number of internal components that will be partially quantized in Chapter 5 and 6. In consequence, we present the LSTM equations that will be needed in subsequent chapters, to clarify the components that will be quantized, and the integration of the quantizers into the LSTM cell. The operation of an LSTM unit is given by the following equations:

$$\begin{aligned}
\mathbf{\Gamma_i}^{(t)} &= \sigma\left(W_{ih}\mathbf{h}^{(t-1)} + W_{ix}\mathbf{x}^{(t)} + \mathbf{b_i}\right), \\
\mathbf{\Gamma_f}^{(t)} &= \sigma\left(W_{fh}\mathbf{h}^{(t-1)} + W_{fx}\mathbf{x}^{(t)} + \mathbf{b_f}\right), \\
\mathbf{\Gamma_o}^{(t)} &= \sigma\left(W_{oh}\mathbf{h}^{(t-1)} + W_{ox}\mathbf{x}^{(t)} + \mathbf{b_o}\right), \\
\tilde{\mathbf{c}}^{(t)} &= \tanh\left(W_{ch}\mathbf{h}^{(t-1)} + W_{cx}\mathbf{x}^{(t)} + \mathbf{b_c}\right), \\
\mathbf{c}^{(t)} &= \mathbf{\Gamma_i}^{(t)} \odot \tilde{\mathbf{c}}^{(t)} + \mathbf{\Gamma_f}^{(t)} \odot \mathbf{c}^{(t-1)}, \\
\mathbf{h}^{(t)} &= \mathbf{\Gamma_o}^{(t)} \odot \tanh(\mathbf{c}^{(t)}),
\end{aligned} \tag{3.7}$$

where $W_{ih}, W_{fh}, W_{oh}, W_{ch} \in \mathbb{R}^{m_h \times m_h}$, and $W_{ix}, W_{fx}, W_{ox}, W_{cx} \in \mathbb{R}^{m_h \times m_x}$ are weight matrices, $\mathbf{b_i}, \mathbf{b_f}, \mathbf{b_o}, \mathbf{b_c} \in \mathbb{R}^{m_h}$ are biases, σ is the sigmoid activation, and $\odot$ is the Hadamard product. The equations are iterated for $t \in \{1, 2, \cdots, T\}$ to obtain the hidden state $h^{(T)}$, that is considered to be the output of the network.

The BiLSTM is a model with two parallel LSTMs, processing the "memory" forward and backward in time [47]. The output of the NN at each time step is a function of the past and future input values. This makes BiLSTM suitable for the equalization of ISI in the communication signals.

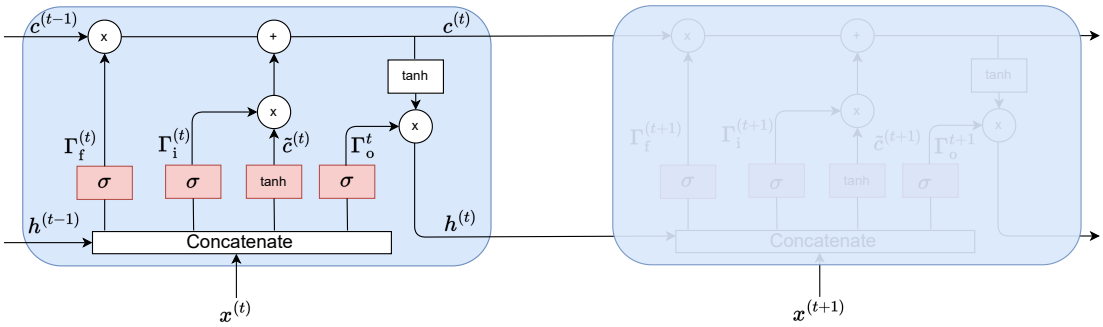


Figure 3.4: Schematic of LSTM Layer

3.3 Training Neural Networks

3.3.1 Stochastic Gradient Descent

Recall that a NN takes an input $\mathbf{x}$ and produces an output $\mathbf{y}_n$, for a given weight vector $\boldsymbol{\theta} \in \mathbf{R}^L$

$$\mathbf{y}_n = h(\mathbf{x}, \boldsymbol{\theta}). \tag{3.8}$$

The parameter $\boldsymbol{\theta}$ is the set of all weights and biases of the NN collected in a vector. For simplicity, we assume that the biases are zero.

The output of the NN $\mathbf{y}_n$ is compared to the desired output $\mathbf{y}$ using an individual loss function $\ell(\mathbf{y}_n, \mathbf{y})$ that measures the error between the two. The task of the training algorithm is to find a set of weights with a small average loss

$$L_S(\boldsymbol{\theta}) = \frac{1}{2n} \sum_{i=1}^{n} l\Big(h(\mathbf{x}^{(i)}, \boldsymbol{\theta}), \mathbf{y}^{(i)}\Big), \tag{3.9}$$

where S is the set of training examples $(\mathbf{x}^{(i)}, \mathbf{y}^{(i)})$, $i = 1, 2, \cdots, n$. The notation $L_S(\boldsymbol{\theta})$ is a shorthand for $L_S(h(\mathbf{x}, \boldsymbol{\theta}))$ introduced earlier. The training algorithm thus minimizes the loss function over the weights

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^l} L_S(\boldsymbol{\theta}). \tag{3.10}$$

This can be done using the gradient descent, as follows.

Recall that the gradient of $L_S(\boldsymbol{\theta})$ is the vector of the partial derivatives with respect to the weights

$$\nabla L_S(\boldsymbol{\theta}) = \left(\frac{\partial L}{\partial w_1}, \cdots, \frac{\partial L}{\partial w_L} \right)^T. \tag{3.11}$$

If the weight w_i changes by a small amount Δw_i, the total variation in the loss function

can be approximated to the first order as

$$\begin{aligned} \Delta L_S(\boldsymbol{\theta}) &\approx \frac{\partial L}{\partial w_1}\Delta w_1 + \cdots + \frac{\partial L}{\partial w_m}\Delta w_m \\ &= \big(\nabla L_S(\boldsymbol{\theta})\big)^T \Delta\boldsymbol{\theta}, \end{aligned} \tag{3.12}$$

where $\Delta\boldsymbol{\theta} = (\Delta w_1, \cdots, \Delta w_L)^T$. Suppose that we select $\Delta\boldsymbol{\theta}$ proportional to the negative of the gradient vector

$$\Delta\boldsymbol{\theta} = -\eta \nabla L_S(\boldsymbol{\theta}). \tag{3.13}$$

The variable η is a small positive value referred to as the *learning rate*. Then,

$$\Delta L_S(\boldsymbol{\theta}) \approx -\eta ||\nabla L_S(\boldsymbol{\theta})||^2. \tag{3.14}$$

Since the squared magnitude of the gradient vector $||\nabla L_S(\boldsymbol{\theta})||^2$ is always non-negative, from (3.14) we obtain that $\Delta L_S(\boldsymbol{\theta}) \leq 0$. Thus, by updating the weights based on (3.13), the value of the cost function cannot increase. This suggest the steepest descent update equation for the weights

$$\boldsymbol{\theta} \mapsto \boldsymbol{\theta} - \eta \nabla L_S(\boldsymbol{\theta}). \tag{3.15}$$

Equation (3.15) is the gradient descent update rule. By continuing the weights update iteratively, we can progressively decrease the value of the loss function until we reach a local minimum.

This simple derivation is based on the approximation in (3.12). However, the gradient descent can be made precise using a more sophisticated analysis.

The training of the NN using the gradient descent requires the gradient $\nabla L_S(\boldsymbol{\theta})$. The backpropagation algorithm is used for calculating this gradient, which we will present in the next section. The NN takes an input example in the training data set and produces the corresponding output, which is then compared to the correct output in the data set using a loss function. The error between the two is then updated backward in the network using the backpropagation. This algorithm accounts for the contribution of each neuron to the error using the chain rule, and outputs the gradient of the loss function with respect to the weights of each neuron. Finally, the weights are updated using the gradient descent update rule. This process is repeated for a sequence of epochs until the loss is sufficiently small.

The backpropagation algorithm can be expensive for large NN with many neurons

and weights, or large data sets. The stochastic gradient descent simplifies the training by approximating the loss function (3.9). Here, the averaging in (3.9) over all training examples is replaced with averaging over a much smaller randomly selected batch set. The batch size is a hyper-parameter that is adjusted by try and error.

3.3.2 Gradient Calculation using Backpropagation

Although the backpropagation algorithm was invented in the 1970s [102], its significance was not widely recognized until the seminal papers of Rumelhart, Hinton and Williams in 1980s [98, 102]. They showed that the multi-layer NNs can be trained using the back-propagation in experiments, faster than the earlier methods of learning. It thus became possible to solve problems with NNs that were previously unsolvable. In the following, we briefly review the back-propagation algorithm.

We consider the setup introduced earlier, with a NN $\mathbf{y}_n = h(\mathbf{x}^{(i)}, \boldsymbol{\theta})$, a data set S and a loss function $L_S(\boldsymbol{\theta})$. The goal of the backpropagation is to calculate the gradient $\nabla L_S(\boldsymbol{\theta})$, *i.e.,* the partial derivatives of the loss function with respect to the weighs $\frac{\partial L_S(\boldsymbol{\theta})}{\partial \theta_i}$. We require two assumptions on the structure of the loss function.

Assumptions. The first assumption is that the overall loss function is a sum of the individual loss functions for each training example over the training data set, *i.e.,*

$$L_S(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} L_i, \tag{3.16}$$

where $L_i := L(\mathbf{x}^{(i)}, \mathbf{y}^{(i)}, \boldsymbol{\theta})$ is an individual loss function depending only on the i^{th} training example (not the whole S). The necessity for this assumption stems from the fact that the backpropagation algorithm described below calculates the partial derivatives of the individual loss function with respect to the weights for a single training example, *i.e.,* it calculates ∇L_i. By computing these partial derivatives for all training examples in S (or a batch set) and averaging them, we obtain the gradient of the overall loss

$$\nabla L_S(\boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \nabla L_i. \tag{3.17}$$

The second assumption is that loss function should be expressed as a function of the NN's outputs (not the hidden activations as well).

In the remaining part of this section, we consider the mean square error (MSE) loss

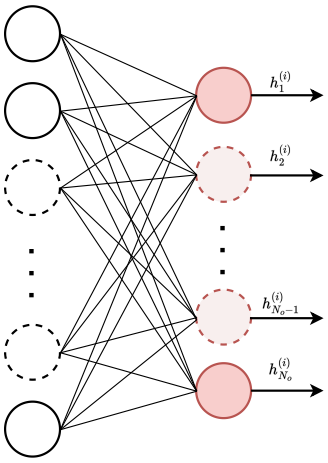


Figure 3.5: The output layer of a NN.

function

$$L_S(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n} ||h(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - \mathbf{y}^{(i)}||^2, \tag{3.18}$$

where the variables were defined in the previous section. Clearly, $L_S(\boldsymbol{\theta})$ is non-negative, and small if the model fits well the training data, *i.e.,* $\mathbf{y}^{(i)} \approx h(\mathbf{x}^{(i)}, \boldsymbol{\theta})$, for most i. The MSE is one of the most widely used loss functions in machine learning.

It can be verified that the MSE satisfies the first assumption, with the following choice of the individual loss function

$$L_i := \frac{1}{2}||h(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - \mathbf{y}^{(i)}||^2. \tag{3.19}$$

The second assumption is also satisfied by the MSE loss. Let $\mathbf{y}^{(i)} = (y_1^{(i)}, \cdots, y_{N_o}^{(i)})$ be the vector of the output labels for the i-th training example, and $h(\mathbf{x}^{(i)}, \boldsymbol{\theta}) = (h_1^{(i)}, \cdots, h_{N_o}^{(i)})$ the output of the NN, where N_o is the number of output neurons. The loss function is a double sum

$$\begin{aligned} L_S(\boldsymbol{\theta}) &= \frac{1}{2n}\sum_{i=1}^{n} ||h(\mathbf{x}^{(i)}, \boldsymbol{\theta}) - \mathbf{y}^{(i)}||^2 \\ &= \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{N_o} |h_j^{(i)} - y_j^{(i)}|^2 \\ &= \sum_{j=1}^{N_o}\left[\frac{1}{2n}\sum_{i=1}^{n} |h_j^{(i)} - y_j^{(i)}|^2\right]. \end{aligned}$$

It follows that $L_S(\boldsymbol{\theta})$ is a sum over scalar terms for each component of the output given

by the term inside the brackets.

Back-propagation algorithm. The il neuron refers to the neuron i (in a given ordering) in layer l. The pre-activation input of this neuron is denoted by I_i^l, so that the corresponding output is $\sigma(I_i^l)$, where σ is the activation function. Denote the loss at one example $(\mathbf{x}, \mathbf{y})$ by R. To compute the gradient element $\partial R/\partial\theta_i$, we introduce an intermediate term δ_i^l that denotes the contribution of the il neuron to the gradient of the individual loss function:

$$\delta_i^l := \frac{\partial R}{\partial I_i^l}. \tag{3.20}$$

In other words, this term represents the change in the loss function when a small disturbance is added to the input of the il neuron.

The backpropagation works by induction, propagating the error from the output of the NN to its input.

The initial step of the induction at the output layer $l = L$. The NN outputs a vector with entries $\sigma(I_i^L)$. The error can be written as

$$\delta_i^L := \frac{\partial R}{\partial I_i^L} = \left(\frac{\partial R}{\partial \sigma(I_i^L)}\right)\left(\frac{\partial \sigma(I_i^L)}{\partial I_i^L}\right). \tag{3.21}$$

The two terms in the right hand side of the above equation can be computed easily. The quantity I_i^L is calculated by the network operating on the input $\mathbf{x}$. The term $\frac{\partial \sigma(I_i^L)}{\partial I_i^L} := \sigma'(I_i^l)$ is the derivative of the activation function that can be computed with minimal effort. For example, for a sigmoid

$$\sigma'(I_i^L) = \sigma(I_i^L)(1 - \sigma(I_i^L)).$$

The term $\frac{\partial R}{\partial \sigma(I_i^L)}$ depends on the cost function. For instance, if we use the MSE loss, then,

$$\frac{\partial R}{\partial \sigma(I_i^L)} = \sigma(I_i^L) - y_i, \tag{3.22}$$

where $y_i := [\mathbf{y}]_i$.

The induction step at layer $l + 1 < L$. Consider the layer $l + 1$ with the weight matrix W^{l+1} and the bias vector $\mathbf{b}^{l+1}$. The error for layer l can be written using the chain rule

$$\delta_i^l := \frac{\partial R}{\partial I_i^l} = \sum_{k \in \mathcal{I}_{il}} \left(\frac{\partial R}{\partial I_k^{l+1}}\right)\left(\frac{\partial I_k^{l+1}}{\partial I_i^l}\right), \tag{3.23}$$

where $\mathcal{I}_{il}$ is the index of the neurons in layer $l+1$ connected to the il neuron in layer l. Note that $\frac{\partial R}{\partial I_k^{l+1}} = \delta_k^{l+1}$. Furthermore,

$$I_k^{l+1} = \sum_m w_{km}^{l+1} \sigma(I_m^l) + b_k^{l+1}, \tag{3.24}$$

where $w_{km}^{l+1} = [W^{l+1}]_{km}$ and $b_k^l = [\mathbf{b}^l]_k$. Differentiating with respect to I_i^l,

$$\frac{\partial I_k^{l+1}}{\partial I_i^l} = w_{ki}^{l+1} \sigma'(I_i^l). \tag{3.25}$$

Thus, (3.23) simplifies to

$$\delta_i^l = \sum_{k \in \mathcal{I}_{il}} w_{ki}^{l+1} \delta_k^{l+1} \sigma'(I_i^l). \tag{3.26}$$

It follows that the errors δ_i^l in layer l can be obtained from the errors δ_i^{l+1} in layer $l+1$ using the recursive relation (3.26).

The gradient of the loss function with respect to the weights is

$$\begin{aligned} \frac{\partial R}{\partial w_{ik}^l} &= \left(\frac{\partial R}{\partial I_i^l}\right)\left(\frac{\partial I_i^l}{\partial w_{ik}^l}\right). \\ &= \sigma(I_k^{l-1})\delta_i^l, \end{aligned}$$

where we used (3.24).

Likewise, the gradient of the loss function with respect to the bias is

$$\begin{aligned} \frac{\partial R}{\partial b_i^l} &= \left(\frac{\partial R}{\partial I_i^l}\right)\left(\frac{\partial I_i^l}{\partial b_i^l}\right). \\ &= \delta_i^l, \end{aligned}$$

where $\frac{\partial I_i^l}{\partial b_i^l} = 1$ from (3.24).

The gradient of the loss function is calculated by iterating the backpropagation equations for $l = L, L-1, \cdots, 1$. The errors are computed backward in layers starting from the output layer.

3.4 Application of the Neural Networks

Deep learning has emerged as a powerful tool with a wide range of applications in different fields. The ability of the NNs to learn complex patterns in data and make accurate predictions has revolutionized the computer vision [69]. Training large NNs on high-resolution

images used to be expensive in the past. However, the advances in hardware, particularly the advent of the Graphical Processing Unit (GPU) and AI accelerators, paved the way for training large models, thus leveraging the potential of the NNs in accurate image classification and object recognition.

NNs are utilized in conjunction with Hidden Markov Model (HMM) to improve the speech recognition systems [90]. In conventional HMMs, a simple Gaussian distribution is used to model the connection between the acoustic features and the corresponding phonetic units. However, by integrating a NN into the system, more complex nonlinear mappings can be captured with improved accuracy in speech recognition.

Anomaly detection refers to the process of identifying data points or patterns in a data set that deviate from the expected behavior [17]. One approach to anomaly detection is based on the dimensionality reduction. There is often redundancy in data set, and correlations among the features in the input data. The aim is to identify a subspace of the domain set $\mathcal{X}$ that captures most of the features in the data relevant to the classifier. The data is projected onto this subspace, and instances with significant reconstruction error are identified as anomalies. The principal component analysis (PCA) is a dimensionality reduction method often used in machine learning. However, the PCA is governed by a linear transformation which fails to capture the complex nonlinear correlations between the features. The NNs, on the other hand, can account for such complex correlations, and may find better subspaces.

NNs have found application in the wireless communication as well [43]. There, they have been used for signal processing, channel estimation, modulation and demodulation, resource allocation, interference cancellation, and optimization. One area where NNs have been particularly effective is channel modeling [126]. By training on large datasets of the channel measurements, NNs can learn to accurately model the characteristics of the wireless channels. This in turn facilitates more efficient equalization, resource allocation and interference management in the wireless communication systems.

CHAPTER 4

Neural Networks for Nonlinearity Mitigation

The subject of machine learning provides statistical signal processing tools that can be used for equalization of the fiber transmission effects [109]. The machine learning-based equalizers might potentially require fewer computational steps than the DBP. Furthermore, these equalizers can be frequently re-trained, making them suitable for adaptive equalization, *e.g.,* in reconfigurable fiber-optic transmission links.

In this chapter, we review the use of NNs for equalization in optical fiber communication. We propose two models for low-complexity nonlinearity mitigation in dual-polarization transmission, that are integrated into the linear DSP chain. Lastly, we compare the Q-factor and complexity of our models, DBP and the linear equalizer.

4.1 Equalization in Optical Fiber With Neural Networks

The use of NNs in data communications dates back to few decades ago. The MLPs were considered a promising approach for the compensation of the nonlinear impairments in wireless communications [16, 96]. The application of the NNs for compensating the impairments in optical fiber was studied in [63]. In this paper, a NN-based nonlinear equalization technique was proposed for mitigating the nonlinearities in coherent optical orthogonal frequency division multiplexing. Compared to an inverse Volterra series transfer function, the NN achieved a 3 dB increase in the Q-factor in a 16-QAM 80 Gb/s 1000-km transmission link. There has been since a growing number of papers on this topic, following the explosion of the interest in deep learning in the past decade.

The NNs used for the nonlinearity mitigation can be divided into two classes. The model-driven NNs have an architecture that is based on an existing equalizer such DBP,

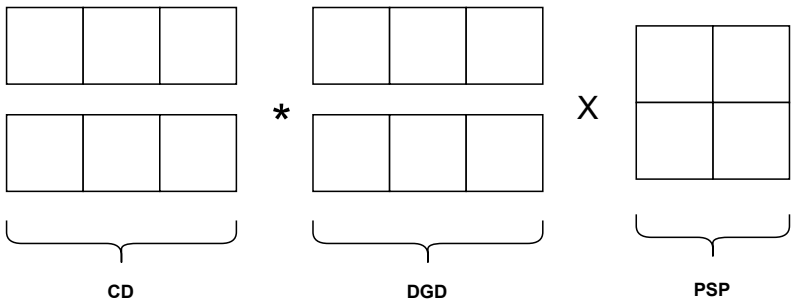


Figure 4.1: Linear layer in learned digital back propagation (LDBP)-PMD, $*$ is the convolution operation.

with the parameters learned from the training data. In model-agnostic NNs, the model acts as a black box and does not require any knowledge of the channel model.

4.1.1 Model-driven Neural Networks

An example of a model-driven NN is the learned digital back propagation (LDBP) proposed in [51]. Here, the authors exploit the fact that the SSFM has the functional form of a NN, and proposed an architecture that imitates the DBP. LDBP has a linear layer to compensate for the CD, and a nonlinear phase activation function to mitigate the nonlinearity. These layers are concatenated a number of times, until the BER is sufficiently low. The authors of [51] report that the LDBP reduces the computational complexity compared to the conventional DBP.

The input output of the LDBP with L layers can be expressed as

$$\mathbf{y} = W^{(L)}\Phi(W^{(L-1)}\cdots\Phi(W^{(1)}(\mathbf{x}))), \tag{4.1}$$

where $\mathbf{x} \in \mathbb{C}^m$ is the input of the NN, which is the signal at the output of the fiber channel typically sampled at at least 2 samples/symbol, and $\mathbf{y} \in \mathbb{C}^m$ is the output of the NN which is the equalized signal before demodulation. Further, $W^{(k)}$ is the weight matrix of the layer k, and $\Phi(.)$ is the activation function.

The model in (4.1) describes a general dense NN. In LDBP, the weight matrix has the form $W^{(i)} = D^H \mathrm{diag}(\widehat{\mathbf{h}})D$, where D is the discrete Fourier transform matrix and $\widehat{\mathbf{h}}$ is the CD filter in the frequency domain with trainable dispersion coefficients. The activation function $\Phi(.)$ introduces a nonlinear phase term, similar to that in DBP [51]. In the single-polarization model, the activation function acts component-wise, and $\Phi(x) = x\exp(j|x|^2)$.

The LDBP is extended from the single- to dual-polarization transmission in [14]. In

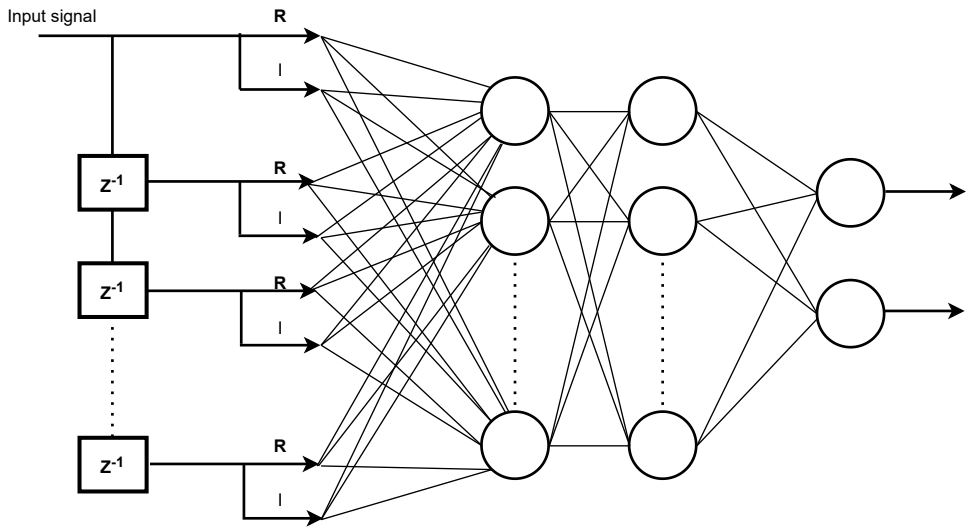


Figure 4.2: The NN nonlinear equalizer in [109].

this paper, a model dubbed LDBP-PMD is proposed for polarization-multiplexed systems, by parameterizing the SSFM for the Manakov-PMD equation. The NN applies filters to dual-polarized signals in a distributed fashion. The linear layer of the LDBP-PMD is decomposed into three parts, shown in Fig. 4.1:

- Two complex-valued symmetric filters, to compensate for CD
- Two real-valued asymmetric filters to mitigate the DGD filters
- A complex-valued 2×2 matrix, to account for the principal state polarization (PSP)-rotating Jones matrix.

LDBP naturally achieves good performance, since the NN architecture is tailored to the channel. However, the NN should be initialized carefully, sometimes around the DBP parameters, in order to converge to a good solution. Moreover, training the complex-valued LDBP can be slightly cumbersome. In this thesis, we do not use model-driven NNs such as LDBPs. The reader is referred to [14] and [51] for further details.

4.1.2 Model-agnostic Neural Networks

The model agnostic NNs have generic architectures that do not depend on the channel. A dynamic model-agnostic NN for nonlinear equalization in long-haul dual-polarization fiber transmission is presented in [109]. The NN is applied after the CD compensation. The network architecture is shown in Fig. 4.2, taking the real and imaginary parts of the signal samples as the input. To take into account the channel memory, delay blocks are

introduced. Thus, to equalize a given symbol at the RX, a number of neighbor symbols in a received symbol stream are required. The size of the input layer is $2(N_d + 1)$, where N_d is the number of delay blocks. The network has two hidden layers, and an output layer of two neurons per polarization, one for the real and one for the imaginary part of the symbol. The neurons in the hidden layer have tangent hyperbolic (tanh) or sigmoid activation function, whereas the neurons of the output layer have no activation.

A fully-connected NN for the nonlinearity mitigation in a short-haul 166-QAM dual-polarization transmission experiment is considered in [15]. The network architecture consists of an input layer that takes the symbols from the x and y polarizations, two hidden dense layers with tanh activation function, and an output layer with 2 neurons that produce one symbol for each polarization. The conventional DSP is applied at the receiver, and the NN is positioned as a component in the digital coherent receiver in two different locations in the DSP chain: once after the MIMO equalization, and once after the CPE and before the symbol detection. The improvements over DBP appear to be small in both cases.

In [110], the authors explore convolutional neural network (CNN)s to mitigate nonlinear distortions in a 16-QAM 3200 km 11x400-Gb/s WDM fiber-optic transmission link. Their main focus is reducing the algorithmic complexity. To achieve this, the authors initialize the weights using a filter that is pre-trained on a single-layer CNN. Additionally, they use an improved activation function that accounts for the nonlinear interactions between the neighbor symbols. To enhance the learning efficiency, they adopt a layer-wise training approach, followed by the joint optimization of all the weights in the multi-layer network through further training. It is shown that the convolutional model can fully compensate the CD, which is expected since dispersion can be expressed by a convolution.

The use of LSTMs to compensate for the fiber nonlinearities in coherent optical transmission is investigated in [30]. The authors perform numerical simulations in single-channel and WDM polarization-multiplexed fiber transmission over the C- and O-band. In order to determine the performance and complexity limits of the LSTM-based receivers, the authors conduct a comprehensive analysis of the impact of the of the number of hidden units, and the number of symbols required for training which is related to the channel memory. The results show that LSTM is comparable to DBP in mitigating the intra-channel effects, and outperforms DBP when inter-channel effects are present. It is shown that training is tolerant to the changes in the signal power and the modulation format of the neighbor

WDM channels, provided that the model is trained in a worst-case scenario where the nonlinear effects are strongest. Lastly, the complexity analysis in [30] indicates that the LSTM could compete with DBP, especially in long distance communication with small accumulated dispersion.

Another relevant paper on the NN-based nonlinear equalization in coherent optical fiber communication is [42]. This work considers an experiment in which the transmission is dominated by the Kerr nonlinearity and component imperfections. The paper compares several NN architectures for equalization, including models that combine a convolutional layer with an LSTM layer or a MLP. The study finds that a convolutional layer in combination with an LSTM layer is the best performing NN among the studied models, for the case that the computational complexity is high. However, when the complexity is low, the best performing structure turned out to be an MLP. This behavior can perhaps be explained by the fact that advanced architectures such as the LSTM include several complex components, while the MLP uses only basic summation and activation functions.

We note that, unlike DBP, machine learning NNs can address the equalization and demodulation in one step. This is achieved by mapping the baseband signal to a latent space learned from a training data set, followed by classification or regression.

4.2 Two Proposed Models for Nonlinearity Mitigation

We propose two models for the nonlinearity mitigation in dual-polarization optical fiber transmission. Due to the constrains of the practical systems, we restrict to low-complexity architectures.

4.2.1 Convolutional-dense Equalizer

The first model is a combination of a convolutional and a dense layer. The architecture of the proposed NN is shown in Fig. 4.3. The four real-valued symbols of the x and y polarizations after the digital coherent receiver over T time steps are denoted by the vectors $\Re(\tilde{\mathbf{s}}_x)$, $\Im(\tilde{\mathbf{s}}_x)$, $\Re(\tilde{\mathbf{s}}_y)$ and $\Im(\tilde{\mathbf{s}}_y)$. The resulting array of shape $(T, 4)$ is fed to the NN. The corresponding symbols at the output of the NN are $\Re(\hat{\mathbf{s}}_x)$, $\Im(\hat{\mathbf{s}}_x)$, $\Re(\hat{\mathbf{s}}_y)$ and $\Im(\hat{\mathbf{s}}_y)$, respectively. The NN operates in a sliding-window fashion: as the vector at the input of the NN is shifted forward two steps in time, two complex symbols are produced. Thus, T is arbitrary.

The model consists of a cascade of three small layers. The first layer includes two

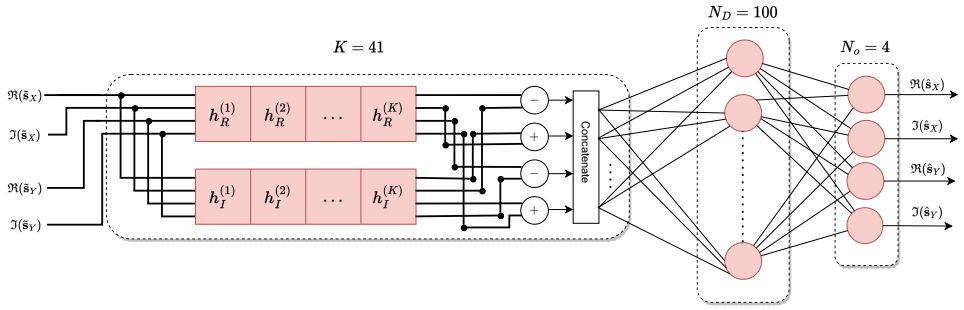


Figure 4.3: The convolutional-dense model. The input is the linearly-equalized symbols $\tilde{s}_x$ and $\tilde{s}_y$, and the output is the fully-equalized symbols $\hat{s}_x$ and $\hat{s}_y$. The convolutional filter taps are indicated by $h_R^{(i)}$ and $h_I^{(i)}$. The activation is tanh is the dense layer, and does not exist in the convolutional and output layer.

parallel real-valued one-dimensional convolutional filters $\left(h_R^{(i)}\right)_{i=1}^{K}$ and $\left(h_I^{(i)}\right)_{i=1}^{K}$ of length $K = 41$ with no activation, for the compensation of CD in the symbols of the x and y polarizations. Each filter is convoluted with each of its input vectors separately, with stride 1 and the same padding. There are total $2K = 82$ real-valued filter taps, far less than in generic convolutional layers used in the literature with large feature maps. The outputs of the convolutional filters are suitably added and subtracted in order to implement 2 complex-valued convolutions from 8 real-valued ones, resulting in four vectors.

The four outputs of the convolutional filters are concatenated in a vector and passed to a fully-connected layer with $N_D = 100$ hidden neurons, and tanh activation. The FC layer processes the two polarizations jointly in order to compensate the cross-pol nonlinear interactions during the propagation. Finally, there is an output layer with $N_o = 4$ neurons, 2 per each polarization symbol, followed by the nearest-neighbor symbol detection.

The NN performs nonlinear regression by minimizing the MSE between its output and the expected output (*i.e.,* the transmitted symbols) in a training data set. The computational complexity of the NN, measured by the number of the floating-point (FP) real multiplications per complex symbol per polarization, is

$$\mathcal{C} = 4K + 2N_D + \left\lceil \frac{N_D N_o}{2K} \right\rceil. \tag{4.2}$$

4.2.2 BiLSTM-dense Equalizer

A BiLSTM-based neural network is used for the second model. The NN takes four vectors of length 21, which correspond to the real and imaginary parts of the x and y symbols. The value 21 is obtained by considering 10 preceding and 10 succeeding symbols for each equalized symbol, to take into account the inter-symbol interference. These four vectors

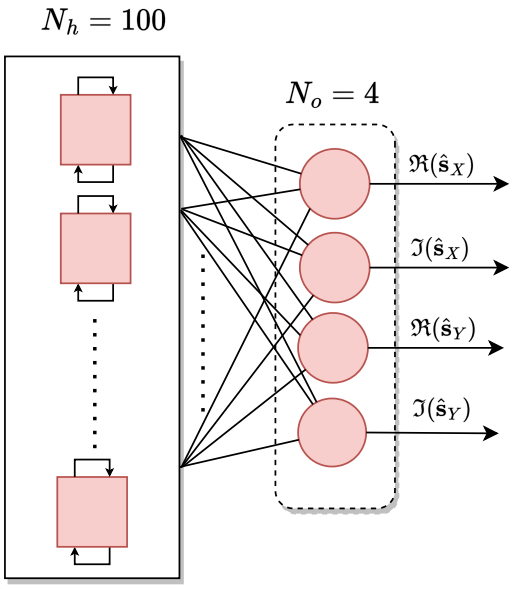


Figure 4.4: The BiLSTM-dense model.

are concatenated into a single vector of length 84. This approach takes into account the nonlinear interactions among the adjacent bits caused by CD.

The concatenated vector is then processed by the BiLSTM layer. This layer has a forward activation function of tanh and a recurrent activation function of sigmoid. It contains 100 neurons and outputs a vector of size 200.

To further process the signal, the output of the BiLSTM layer is fed to a fully-connected layer with no activation and 4 neurons. The BiLSTM-dense architecture is shown in Fig. 4.4.

4.3 Performance Results

Fig. 4.5 shows the block diagram of the transmission experiments considered in this thesis. Three experiments are performed with: Truewave classic (TWC) fiber, single mode fiber (SMF), and LEAF fiber.

At the transmitter (TX), two sequences of bits are generated for the x and y polarizations, that are gray coded to two sequences of complex symbols taking values in a 16-QAM constellation. The two complex-valued symbols are converted to four real sequences corresponding to the I and Q components of the x and y polarizations, shaped with the root raised cosine (RRC) filter with the roll-off factor of 0.1, by an arbitrary wave generator (AWG) at 34.4 GBaud. The AWG contains the digital-to-analog converters (DACs) at 88 Gsamples/s. The output is then amplified using the electrical amplifiers. The electrical

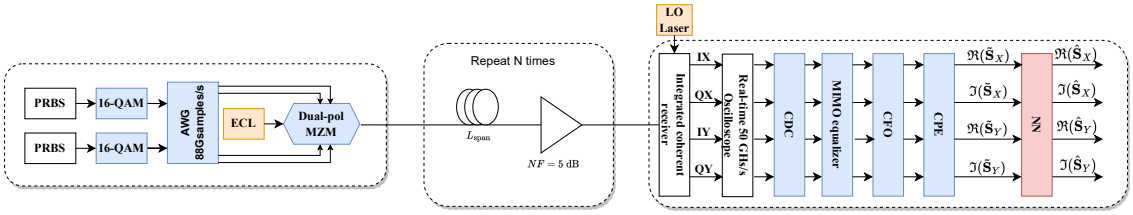


Figure 4.5: The experimental transmission of dual-polarization 16-QAM at a rate of 34.4 GBaud.

signals are converted to optical signals and polarization multiplexed with a Mach-Zehnder modulator (MZM), driven by an external cavity laser (ECL) at wavelength 1.55 μm with the line-width 100 KHz. The optical signal is sent over a fiber link, in three configurations that will be described in the next sections and summarized in Tab. 4.1.

At the receiver RX, the first step is to perform polarization demultiplexing, which separates the signal into two orthogonal polarizations. This is done using an integrated coherent receiver, which converts the optical signal to four electrical signals. These signals are then sampled by analog-to-digital converters (ADCs) at the rate of 50 Gsamples/s. The ADCs have the effective number of bits (ENoB) of 5.

After conversion to the electrical signals, the linear DSP chain is used to compensate for the chromatic dispersion (CD), followed by MIMO equalization with radius directed equalizer, and carrier phase estimation (CPE). The linear DSP chain is shown in Fig. 4.5 and Fig. 2.16, and explained in Section 2.5.

Once the linear DSP is applied, the signal is still subject to the dual-polarization nonlinearities and the distortions introduced by the devices. To mitigate these imperfections, the signal is passed to a low-complexity NN to minimize the impact of the nonlinearities and other distortions. The architecture and training of the NN depends on the experiment, and explained below.

4.3.1 TWC Experiment

In a laboratory, a short-distance optical transmission experiment was conducted using the TWC fiber with a span length of 50 km. The optical signal was sent over a straight-line optical fiber link comprising of 9 spans. To compensate for the fiber loss, an EDFA with a 5 dB noise figure was placed at the end of each span. The fiber channel had a loss of 0.21 dB/km, chromatic dispersion of 5.5 ps/(nm $\cdot$ km) and a nonlinearity parameter of 2.8 $(\text{Watt} \cdot \text{km})^{-1}$. The channel was operated in the nonlinear regime at high powers, considering the low dispersion and high fiber nonlinearity parameter.

	Setup 1	Setup 2	Setup 3
Fiber type	TWC	SMF	NZDSF(LEAF)
Modulation	16-QAM	16-QAM	16-QAM
Baud rate (Gbaud)	34.4	34.4	34.4
α (dB/km)	0.21	0.22	0.19
D (ps/(nm-km))	5.5	18	4
γ $(\text{Watt} \cdot \text{km})^{-1}$	2.8	1.4	2.1
PMD $(\text{ps}/\sqrt{\text{km}})$	0.02	0.08	0.04
Noise figure (dB)	5	5	5
Span number	9	9	17
Span length (km)	50	110	70

Table 4.1: Transmission and channel parameters

The NN is the BiLSTM-dense model shown in Fig. 4.3. The hyper-parameters of this model are the size of the convolutional filters K and the number of hidden neurons N_D. The filters' length is determined by the channel memory, measured in the number of symbols due to the residual dispersion left after the CD compensation. This is estimated to be 40 symbols, through the correlation function of the received symbols after CPE, or performance evaluation. The minimum number of hidden units is 100, below which the performance rapidly drops.

The training set contains 600,000 symbols from a 16-QAM constellation. A test set of 100,000 symbols is used to assess the performance of the NN. Each dataset is measured at a given power, during which the BER may fluctuate in time due to the environmental changes. The symbols on the boundary of the data frame are eliminated to remove the effects of the anomalies. The NN at each power is trained and evaluated with independent datasets of randomly chosen symbols at the same power. The NN is built, trained and evaluated in the Python's TensorFlow library. The loss function is the mean-squared error, and the learning algorithm is the Adam-Optimizer with the learning rate of 0.001.

The performance of the NN is compared with that of DBP and linear equalization. The DBP replaces the CD compensation module, and is applied with single step per span, and 2 samples per symbol. This comparison is done to evaluate the effectiveness of the NN in jointly mitigating the residual chromatic dispersion and Kerr nonlinearity.

Fig. 4.6 compares the Q-factors of the proposed NN and linear DSP with respect to the average power of the transmitted signal. The results demonstrates that the NN offers a Q-factor enhancement of 0.5 dB at -2 dBm, and 2.3 dB at 2 dBm. The improvement results from the mitigation of the cross-pol nonlinearities, as well as the equipment's distortions. The raw data before the linear DSP was not available to add the DBP curve to Fig. 4.6.

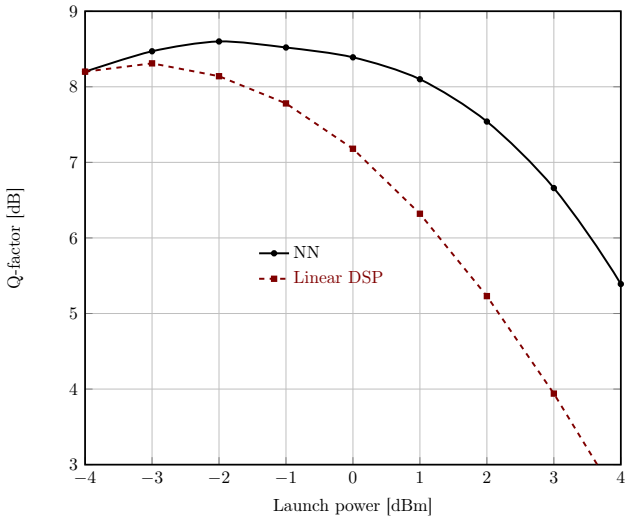


Figure 4.6: Performance of the convolutional-dense equalizer, compared to the linear DSP in the TWC experiment.

However, it is shown in [42] that the Q-factor of the NN is comparable to that of a DBP with 3 steps/span at 2 dBm on the same experimental data set.

4.3.2 SMF Experiment

In this experiment, we consider a long-haul transmission link with 990 km length. The experimental setup is same as that in the TWC case, except that the link is comprised of 9 spans of 110 km length of SMF. The fiber had a loss of 0.22 dB/km, chromatic dispersion of 18 ps/(nm-km) and a nonlinearity parameter of 1.4 $(\text{Watt} \cdot \text{km})^{-1}$.

The NN used in this experiment is the convolutional-dense model depicted in Fig. 4.3. We restrict the memory to 40 symbols, set the convolutional filter length K to 40, and choose $N_D = 100$. These parameters were selected upon a comprehensive analysis of the NN equalizer's effectiveness under varying conditions. To asses the performance of the NN equalizer, we use a training dataset of 200,000 symbols and a test dataset comprising 100,000 symbols.

Based on the data presented in Fig. 4.7, it can be seen that the NN equalizer offers a substantial improvement over the linear DSP. Specifically, at the power level of 2 dBm, the NN delivers a gain of 1.25 dB, while at a power level of 5 dBm, the improvement is estimated to be 2.17 dB. Additionally, the results indicate that the performance of the NN is comparable to the DBP with one step per span. At a power level of 2 dBm, the DBP provides an enhancement of 0.96 dB, while at a power level of 5 dBm, the enhancement is 1.36 dB. This suggests that the NN is a viable alternative to DBP.

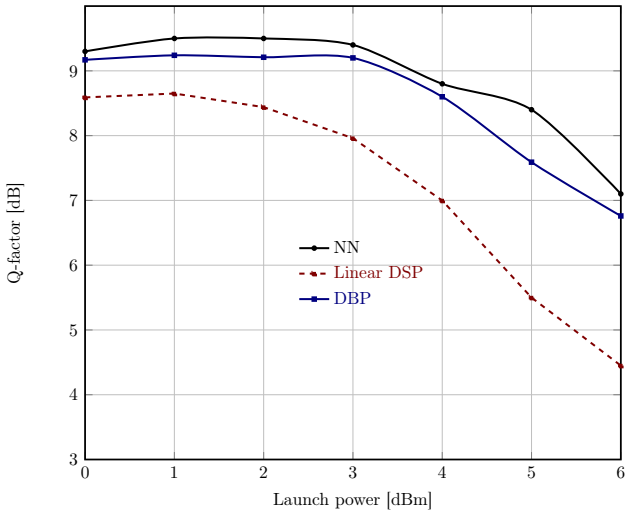


Figure 4.7: Performance of the convolutional-dense equalizer compared to linear DSP and DBP in the SMF experiment.

4.3.3 LEAF Experiment

In this experiment, we consider a long-haul transmission link with 1149 km length. The experimental setup is same as that in the previous two experiments, except that the channel consists of 17 spans of 70 km length of LEAF fiber. This fiber has a loss of 0.19 dB/km, chromatic dispersion of 4 ps/(nm − km) and a nonlinearity parameter of 2.1 $(\text{Watt} \cdot \text{km})^{-1}$.

The two NN architectures used in the previous two experiments are tested on this setup. For the CNN-dense architecture, we simulated the model with the value of $K = 40$ for the convolutional filter length, and $N_D = 100$ for the number of dense units. On the other hand, the BiLSTM architecture in Fig. 4.4 was implemented using an input with a memory size of 20 symbols and 100 hidden units. Each receiver is trained on a dataset consisting of 200,000 symbols, and then evaluated using an additional dataset of 100,000 symbols.

Our analysis of the CNN-Dense architecture in comparison to the linear DSP shown in Fig. 4.8 indicates a significant improvement in performance. Specifically, at the power level of -1 dBm, the CNN-Dense architecture yields an enhancement of 0.78 dBm, while at a power level of 1 dBm, the improvement is even more pronounced at 1.7 dBm. However, when compared to the performance of the DBP, we observed a decrease in performance at high transmission power levels. Specifically, we noted a drop of 0.44 dB at 1 dBm compared to DBP, and this drop became more significant as the transmission power increased.

It is worth noting that we attempted to optimize the hyper-parameters of the CNN-

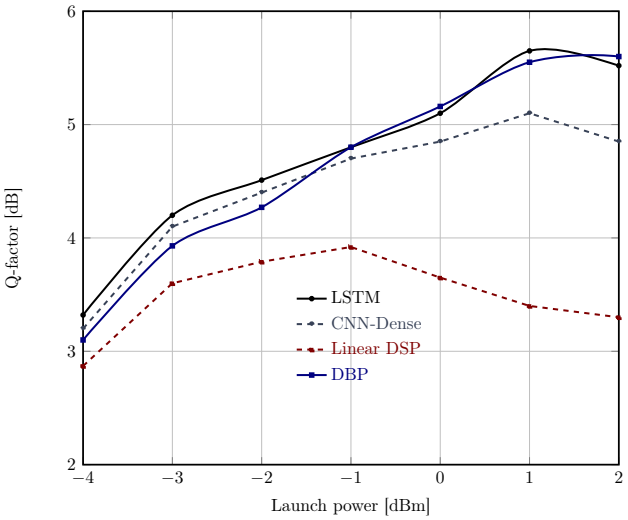


Figure 4.8: Performance of the BiLSTM-dense (NN 1) and convolutional-dense (NN 2) equalizers compared to linear DSP and DBP in the LEAF experiment.

Dense module to enhance its performance, but these efforts did not yield any improvement. This can be attributed to the challenging nature of the transmission setup, as evident in the performance of the linear DSP in Fig. 4.8.

Our evaluation of the BiLSTM based receiver revealed a notable improvement in performance compared to the CNN-Dense architecture. At the power level of -1 dBm, the BiLSTM model yielded a gain of 0.88 dB over linear equalization, while at the power level of 1 dBm, the improvement increased to 2.25 dB. These results are comparable to the performance of the DBP, which provided a gain of 2.15 dB over the linear equalization at 1 dBm.

The enhanced performance of the BiLSTM based receiver can be attributed to its internal memory, which provides an advantage when dealing with the effects of the nonlinearity and dispersion in the transmission medium. However, this improvement comes at the cost of increased complexity. While the CNN-Dense architecture had 16,000 trainable parameters, the BiLSTM has almost 100,000 trainable parameters. Therefore, a trade-off must be made between the complexity and performance when selecting an appropriate receiver architecture for a particular application.

CHAPTER 5

Quantization of Neural Network Equalizers Above 5 bits

The NN equalizers in the optical fiber transmission have to be eventually implemented in ASICs that have limited computational, memory and energy resources. Furthermore, the equalizers should compensates the channel impairments in real-time, requiring low latency. Thus, the training and inference of the model must be optimized for low latency, energy consumption and storage.

The computational complexity and memory requirements of the NNs can often be drastically reduced using the quantization and pruning, with little impact on the prediction accuracy. Different NN quantization schemes have been explored in the literature. In post-training quantization (PTQ), the weights and activations of the NN are quantized after training in full precision. In contrast, in training-aware quantization (TAQ), quantization is integrated in the training algorithm.

Quantization and pruning have been a driver of the high-throughput AI accelerators [8]. However, it has been observed that TAQ below half-precision requires significant tuning of the model. In consequence, much of the recent research on quantization has focused on PTQ due to its simplicity.

In this chapter, we begin by providing an introduction to the quantization in neural networks. We specify metrics for measuring the performance, complexity and storage requirements of the quantized models. Several PTQ and TAQ algorithms are presented for the quantization of the NNs used for the nonlinearity mitigation in fiber-optic transmission experiments described in Chapter 4. This includes a novel companding quantization algorithm that takes advantage of the probability distribution of the weights. We compare the

Q-factor performance and complexity of the PTQ and TAQ, with uniform, non-uniform, fixed- and mixed-precision quantization. Finally, we highlight the limitations of the PTQ and TAQ in quantization below 5 bits, motivating the extreme quantization with a hybrid approach that will be presented in Chapter 6. The material in this chapter is based on the conference papers [26, 28].

5.1 Quantization of Neural Networks

The parameters (weights and biases) of the NN, activations and input data are initially real numbers represented in float 32 (FP32) or float 64 (FP64), described, *e.g.,* in the IEEE 754 standards. The realization of the NN in memory or computationally restricted environments requires that these numbers be represented by fewer number of bits and in different format, *e.g.,* in INT8 format. Thus, the real numbers are quantized in a codebook with a finite set of discrete values

$$\mathcal{W} = \{0, \hat{w}^{(1)}, \cdots, \hat{w}^{(N)}\},$$

where $\hat{w}^{(i)}$ are the quantization symbols. The quantization rate of $\mathcal{W}$ or precision is defined to be $b = \log_2 N$ bits. The zero symbol does not contribute to the rate in some definitions, since it can be obtained via pruning before quantization (the non-zero weights are quantized).

Below, we review a variety of NN quantization schemes that have been proposed in the digital communication and machine learning literature.

5.1.1 Uniform Case

In uniform quantization, the quantization symbols $\hat{w}^{(i)}$ are uniformly placed between a minimum and maximum value. Let w be a full precision parameter anywhere in the NN, (a, c) the smallest interval containing the quantized parameters referred to as the clipping range , $N = |\mathcal{W}| - 1$ and

$$s(a, c, N) = \frac{c - a}{N - 1}.$$

The uniformly quantized weight is represented as in [62]:

$$\hat{w} = \left\lfloor \frac{c(w,a,c) - a)}{s(a,c,N)} \right\rceil s(a,c,N) + a, \tag{5.1}$$

where $\lfloor . \rceil$ is the nearest integer, and

$$c(w,a,c) = \min(\max(w,a),c),$$

is the clipping function. The procedure for selecting the clipping range is called calibration.

Symmetric quantization partitions the clipping range in a symmetric way, *i.e.*, $a = -c$. This approach is easy to implement. However, it is proven to be sub-optimal, if the range is skewed. In this case, asymmetric calibration where $c = \max(w)$ and $a = \min(w)$ provides a better performance. On the other hand, the asymmetric scheme is prone to outliers that unnecessarily increase the clipping range causing performance degradation. One approach to address the problem of the outliers is to select a and c such that the information distance measured by the KL divergence between the unquantized and the quantized values is minimized [82].

5.1.2 Static versus Dynamic

Quantization is said to be of static range if a and c are known and hard-coded a priori in hardware for both weights and activations. The same values are used in training and inference, and for all runs. In contrast, in dynamic range quantization, a and c are computed separately for each component of the network and input. This approach requires real-time computation of the statistics, which increases the algorithm's complexity. However, since the weight span is precisely computed for every particular input, dynamic quantization often results in higher performance.

5.1.3 Non-uniform Case

In nonuniform quantization, the constraint that the quantization levels $w^{(i)}$ are uniformly-spaced is relaxed. The nonuniformly quantized weight can be described as:

$$\hat{w} = w^{(i)}, \text{if } w \in [\Delta_i, \Delta_{i+1}), \tag{5.2}$$

where Δ_i is the i^{th} quantization threshold. The thresholds Δ_i are not uniformly spaced.

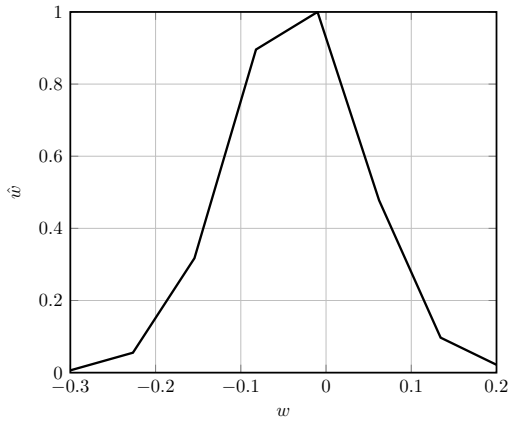


Figure 5.1: The weight density of the dense layer in the NN equalizer. The weights have a skewed bell-shaped distribution, suggesting that uniform quantization is not optimal.

The nonuniform quantization schemes are difficult to deploy on hardware, due to the requirements of iterative clustering techniques [108]. Thus, the majority of the quantization algorithms adopt uniform quantization. However, empirical investigations show that the weight distribution of the dense layers is bell shaped [52], so the nonuniform quantization can provide better compression ratios compared to the uniform schemes.

Power-of-two Quantization

The power of two (PoT) quantization [74] simplifies the implementation by converting multiplications to additions. Here, the quantization codebook is

$$\mathcal{W}(\alpha, b) = \pm\alpha\Big\{0, 2^0, 2^{-1}, \cdots, 2^{-(N/2-1)}\Big\}, \tag{5.3}$$

where $N = 2^b$ and α is stored in FP32, but is applied after the multiply-accumulate (MAC) operations. Hence, the MAC is still in integer addition. The factor α is adjustable.

Note that the multiplication of a PoT number 2^i for some integer i and a finite-bit floating-point number R can be performed efficiently with a bit-wise shift:

$$2^i R = \begin{cases} R, & i = 0, \\ R \ll i, & i > 0, \\ R \gg i, & i < 0, \end{cases} \tag{5.4}$$

where $R \ll i$ (resp. $R \gg i$) means shifting the sequence of bits representing R by i positions to the left (resp. to the right) with zero padding, and converting the result back

to a real number. The bit shift operations have a constant complexity with respect to the bit-width of R, and can be executed in a single clock cycle in the central processing unit (CPU).

Note also that, as the bit-width b is increased by one in (5.3), $\mathcal{W}(\alpha, b+1)$ would be $\mathcal{W}(\alpha, b)$ together with a number of new quantization symbols. This is seen in Fig. 5.2(a)-(b), where the interval $[-2^{-2^{b-1}+1}, 2^{-2^{b-1}+1}]$ is further divided as b is increased from 3 to 4. However, the regions in $[-1, 1]$ outside the above interval remains unchanged as b is increased. This is referred to as the rigid resolution in the PoT quantization.

In the additive PoT (APoT), each quantization symbol is a sum of n PoT values, for some $n \in \mathbb{N}$. Choose a base number of bits b_0 such that $n = b/b_0$ is an integer. Then, the quantization codebook of APoT is

$$\mathcal{W}'(\gamma, b) = \gamma \sum_{i=0}^{n-1} 2^{-i} \mathcal{W}^n(\alpha, b_0) + \beta, \tag{5.5}$$

where γ and β are scale and shift factors in FP32 that are trainable, and the set power is defined component-wise. It can verified that $|\mathcal{W}'| = 2^b$. The shift parameter β allows restricting the quantized weights to unsigned numbers.

Note also that, unlike PoT, as the bit-width b is increased by one in (5.5) in APoT, the quantization symbols in general all change (see Fig. 5.2(c)).

Companding Quantization

Companding quantization has the speed of the uniform quantization, combined with the improved performance of the nonuniform quantization [124]. Companding quantization is a nonuniform technique that involves nonlinearly transforming the data so that a uniform quantizer can be applied. This scheme is proven to perform well if the distribution of the data can be numerically described, or approximated analytically.

A compander is a module composed of a compressor, a uniform quantizer, and an expander. An example of a compander is the μ law, in which the compression part is described for any given input w by:

$$F(w) = \operatorname{sign}(w) \frac{\log(1 + \mu|w|)}{\log(1 + \mu)}, \tag{5.6}$$

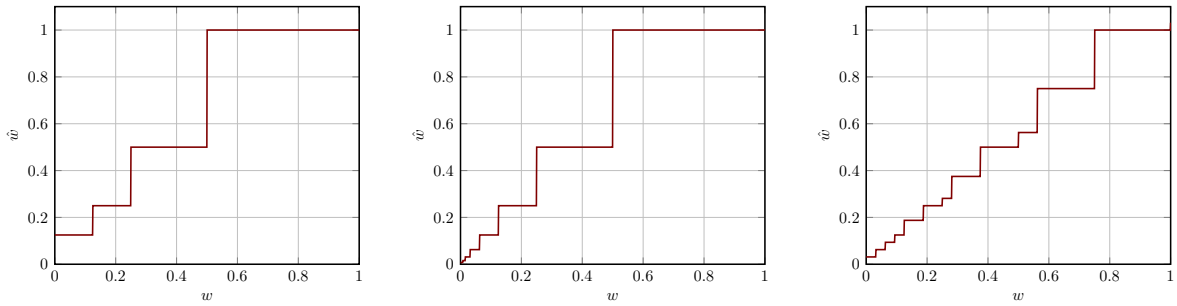


Figure 5.2: The quantizer function of the PoT and APoT quantization. (a) PoT with 3 bits; (b) PoT with 4 bits; (c) APoT with 4 bits.

and the expansion function by

$$F^{-1}(w) = \text{sign}(w)\frac{(1+\mu)^{|w|}-1}{\mu},$$

where μ is defined to be the compression parameter.

Companding quantization has been widely used in digitization, compression and transmission of audio signals. In image classification, the authors of [93] investigated the 2-bit logarithmic companding scalar quantization to compress the weights of a MLP. Through analytical and experimental analysis, they showed that companding-based quantization performs better than the uniform quantization. In a [123], the authors introduced a novel approach to nonuniform quantization that uses a companding scheme. Notably, both the compression and expanding functions are included in the loss function of the NN. They demonstrated that the companding quantization outperforms APoT and uniform quantization when applied to image classification tasks. However, the use of companding quantization in NN has not been much investigated.

5.1.4 Fixed- and mixed-precision

The majority of the quantization schemes consider fixed-precision quantization, where a global bit-width is predefined. However, studies have shown that the optimal bit-width can vary across different layers [134, 75].

In the mixed-precision quantization, different layers, feature maps, channels, weight groups or activations are quantized generally at different rates, as shown in Fig.5.3 [95]. However, it was shown that the search space for the finding the bits is exponential in the number of layers [34]. One approach to determine the bit values is based on the

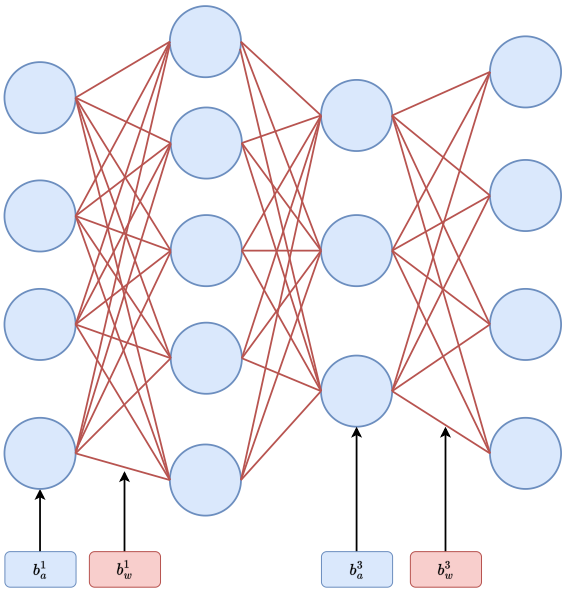


Figure 5.3: Schematic of mixed precision quantization.

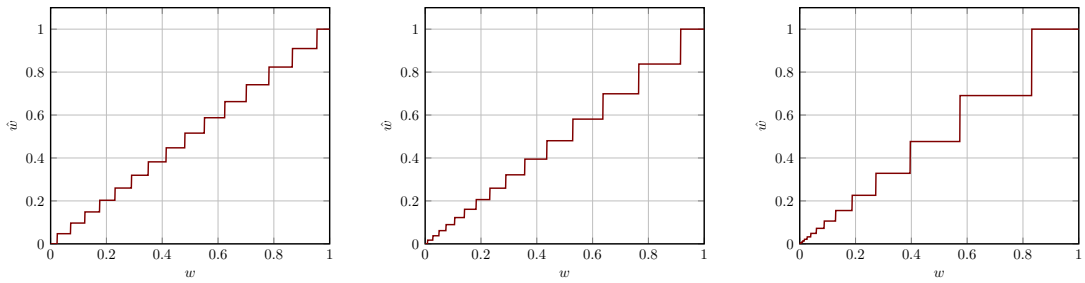
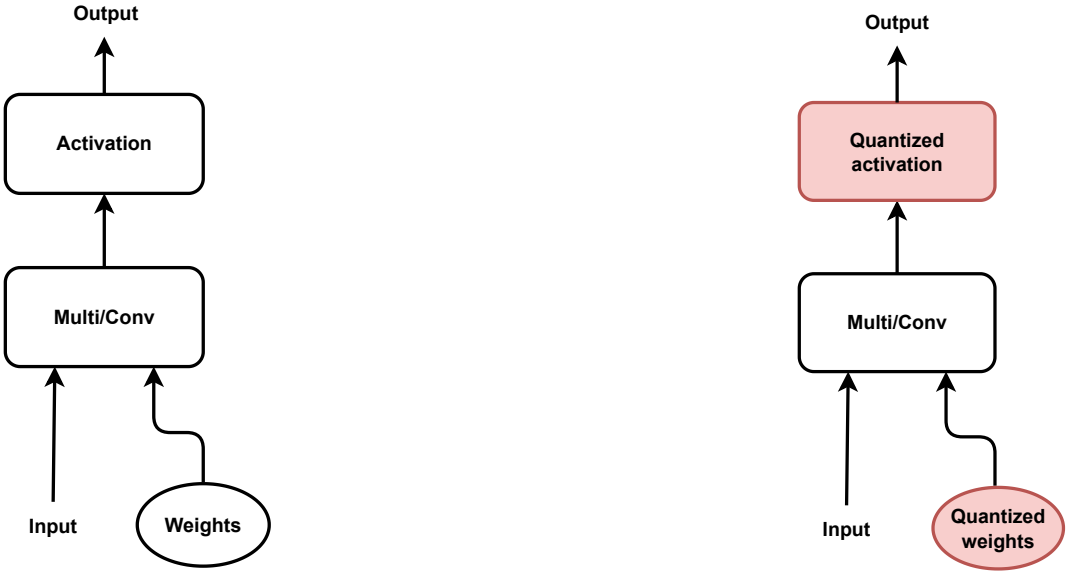


Figure 5.4: Companding quantization at 4 bits for: (a) $\mu = 1$; (b) $\mu = 10$; (c) $\mu = 250$.

second-order sensitivity of the model using the Hessian matrix of the loss function [33]. If the Hessian matrix has a large magnitude for a particular layer, the output of the NN is sensitive to that layer. Consequently, higher bit-width should be assigned to that layer in quantization. In our work the quantization rates are determined depending on the sensitivity of the loss function.

An alternative approach is to explore the quantization space via NAS [116]. However, this approach can be computationally complex, especially in a high-dimensional search space [117].

When a trained NN model is quantized, a perturbation is introduced in the model parameters, resulting in a deviation with respect to the original model operating in floating-point precision. This difference is the quantization noise, which causes a decrease in the accuracy. Below, two different quantization schemes are discussed: PTQ, and TAQ which



(a) Training (b) Inference

Figure 5.5: A diagram explaining PTQ: a) the training of the NN in full precision; b) the inference using quantized values.

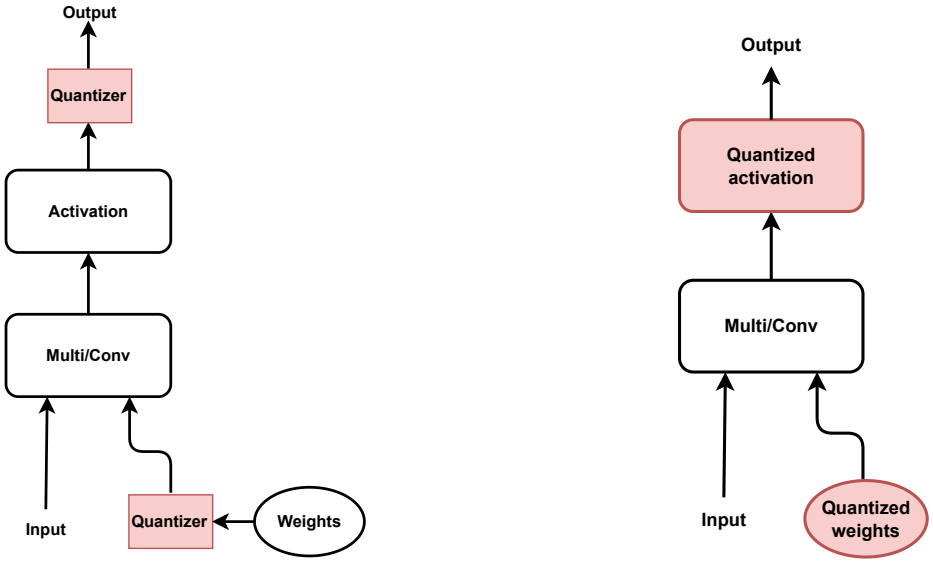
mitigates the quantization noise.

5.1.5 Post-training Quantization

In PTQ, training is performed in full (FP32) or half (FP16) precision. The input tensor, activation outputs, and the resulting weights are then quantized and used in inference [18]. This approach is useful in scenarios where training data is not available (required in TAQ), and requires little to no overheads. However, quantizating below 8 bits can cause a significant degradation in the NN performance [58].

Various approaches have been proposed to mitigate the degradation in performance especially in low bits regimes. Approximating the clipping values analytically from the distribution of the weights, determining the optimal bit-width for each layer of the NN, and correcting the bais in the mean and variance of the quantized weights can reduce the number of bits to 4 while maintaining a good accuracy [7]. It is observed that assigning a floating-point weight to its nearest quantization symbol may not be optimal [89]. The adaptive rounding algorithms can be used to quantize up to 4 bits, while maintaining low degradation in performance [89].

The integration of the PTQ in the NN can be seen in Fig. 5.5. During the training of NN, full precision is utilized, and the quantization noise is not taken into consideration (Fig. 5.5(a)). During the inference, the weights are substituted with their quantized ver-



(a) Training (b) Inference

Figure 5.6: A diagram explaining TAQ: a) the training of the NN in full precision; b) inference using the quantized weights and activations.

sions, and an additional quantizer module is introduced to the activation block to cast the output into low precision (Fig. 5.5(b)).

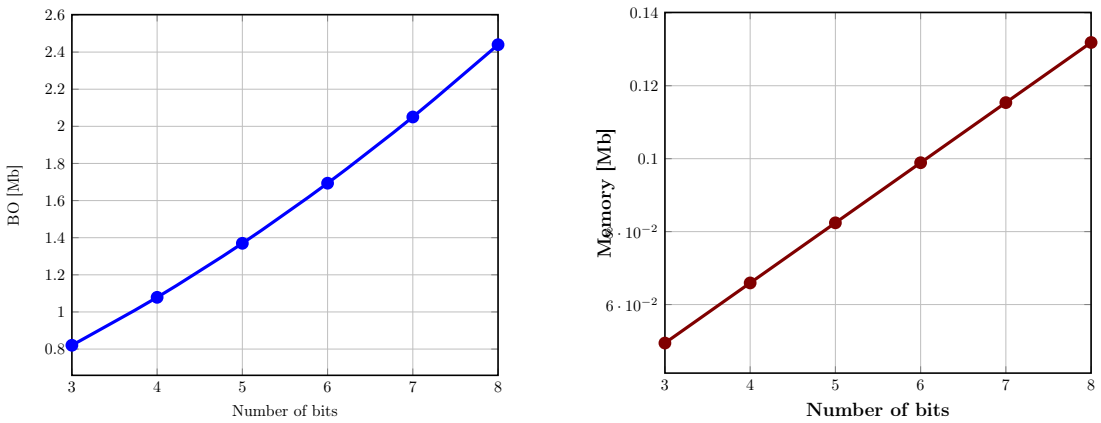
5.1.6 Training-aware Quantization

In TAQ, the quantization and training algorithms are simultaneously developed. This technique usually enhances the prediction accuracy of the model by accounting for the quantization noise during the training. However, learning via the backpropagation of errors in SGD is not possible directly, since the quantizer is a piece-wise flat function with zero derivative almost everywhere.

Straight-through Estimator

The straight-through estimator (STE) is an empirical method that addresses the problem of the zero gradient by modifying the chain rule for differentiation in SGD to ensure a non-zero approximate gradient [127]. The most widely used surrogate for the gradient is the identity function, in which $d\hat{w}/dw \triangleq 1$ [10]. Even though one is not a good approximation of zero, STE works surprisingly well in some models. In this thesis, TAQ in our simulations and figures refers to training with STE.

TAQ typically provides higher prediction accuracy than PTQ when quantizing at low number of bits, at the cost of increased computational and implementation complexity. On



(a) (b)

Figure 5.7: Memory requirements and computational complexity, measured in bit-wise operations (BO), for the convolutional-dense equalizer at varying quantization levels.

the other hand, if the approximation technique is not carefully chosen, TAQ may perform even worse than PTQ [78].

TAQ can be integrated in the training of the NN based on Fig. 5.6. In the training phase of the NN, illustrated in Fig. 5.6(a), the quantizer blocks are positioned after the weights and activations to guarantee that all parameters are represented with low bits. Training can be done from scratch, or from a pre-trained model, with TAQ fine-tuning the result. The latter approach typically yields superior performance with a small number of epochs.

5.1.7 Quantization in Fiber-optic Equalization

With the exception of a few papers, the quantization of the NNs for equalization in optical fiber transmission has largely not been explored. Here, we provide a description of four papers published recently in this area. These papers have shown that it is possible to quantize the weights of the NN equalizers, however, the activations are still in full precision. In contrast, in this thesis, both the weights and activations are quantized, which is actually quite important (see the concluding remarks in this chapter).

In [68], an approach to addressing the nonlinear effects via the over-parameterized NNs that do not require multipliers is proposed. This approach involves representing the weights using a PoT expression, with comparable performance to the that of the original model. Similarly, in [97], an MLP-based nonlinear equalizer is pruned and quantized at 8 bits, with a small degradation in performance.

A recent study [41] reported the performance of PTQ and TAQ for different transmission setups. The results demonstrate that TAQ allows quantization below 6 bits, with minor performance degradation. It is important to note that, as in [68, 97], the activations are not quantized.

Finally, the authors of [54] investigated the effects of the quantizing the complex-valued NN weights. They showed that the network can be quantized with as low as 3 bits and a small degradation in performance. Thus, quantization of the NNs for equalization in optical fiber transmission can significantly reduce the computational complexity with minimal degradation in performance.

5.2 Reduction in the Computational Complexity and Memory

This section introduces metrics for measuring the memory and computational complexity of the two NN equalizers proposed in Chapter 4.2 after quantization.

5.2.1 Multiply-Accumulate

The basic building block of the NNs is the multiply-accumulate (MAC) unit. The computational complexity of a NN can then be determined by the number of bit-wise operations required by the multiply-accumulate (MAC) units.

A MAC unit is a component of a microprocessor or DSP that performs two operations in a single clock cycle: multiplication and accumulation. The unit takes two numbers, multiplies them together, and adds the result to a third number stored in an accumulator register. The accumulator can be pre-loaded with an initial value, and subsequent multiplication operations can be accumulated in the register. The MAC units are used in neural networks for performing the dot products between the weights and activations during the forward pass.

We first calculate the storage requirement of a dot product. Define the bit-width of a scalar to be the number of bits required to store it. Consider the dot product between an input vector of length n whose every element has bit-width b_1 bits, with a weight vector of the same length with per-element bit-width of b_2 bits. The addition of two scalars of bit-width b is a scalar with bit-width $b+1$. The multiplication of two numbers with bit-width b_1 and b_2 bits, respectively, is a number with b_1+b_2 bits. The addition of n numbers

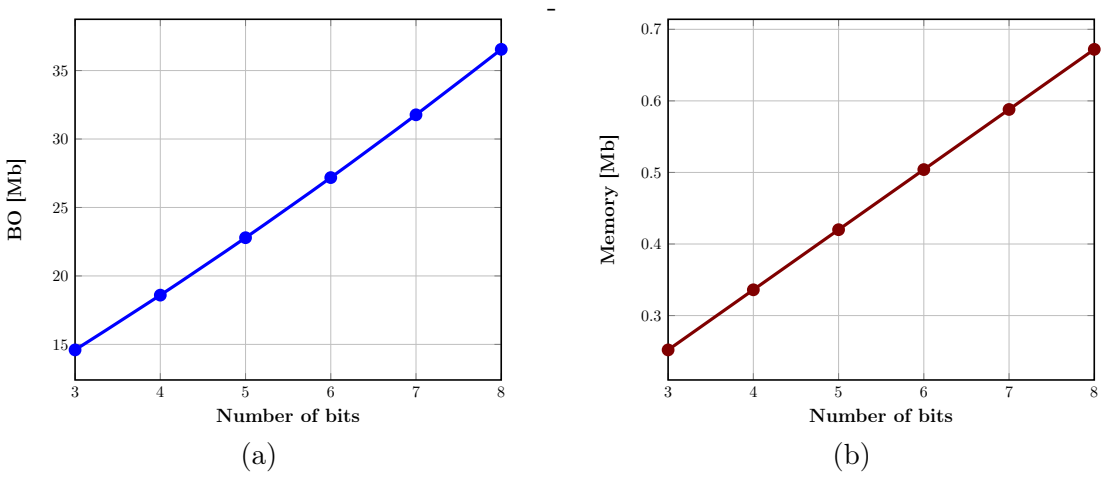


Figure 5.8: Memory requirements and computational complexity, measured in BOs, for the BiLSTM-dense equalizer at varying quantization levels.

can be done in $\log_2 n$ steps by pairwise addition. Thus, the storage requirement of the dot product is $b_1 + b_2 + \log_2(n)$ bits.

The computational complexity of the dot product measured in the bit-wise operations (BO) can be counted similarly. The dot product between two input vectors of length n requires n scalar multiplications and $n-1$ scalar additions. The multiplication of two numbers respectively with bit-width b_1 and b_2 bits takes $b_1 b_2$ elementary bitwise operations. Each addition takes no more than the length of the entire accumulator. Therefore, the dot product requires at most

$$\mathrm{BO}_{\mathrm{dot}} = (nb_1b_2 + (n-1)(b_1 + b_2 + \log_2(n))). \tag{5.7}$$

elementary bit-wise operations.

5.2.2 Dense layers

A dot product is performed by each neuron in a dense layer. As a result, the total number of dot products required to compute the layer is equal to the number of neurons. The BO required to implement a dense layer with input size n_i and n_d neurons is,

$$\mathrm{BO}_{\mathrm{Dense}} = n_d[n_i b_w b_i + (n_i - 1)(b_w + b_i + \log_2(n_i))], \tag{5.8}$$

where b_w and b_i are the bit-widths for the weights and input respectively.

5.2.3 Convolution layers

A single convolution is a dot product. The total number of dot products needed to compute a convolutional layer equals to the number of output features of the layer. For a 1-D convolutional layer with an output size the same as the input size n_i, *i.e.,* with the same padding, and a kernel length m, BOs is

$$\mathrm{BO}_{\mathrm{Conv}} = n_i[mb_kb_i + (m-1)(b_k + b_i + \log_2(m)], \tag{5.9}$$

where b_i and b_k are respectively the input and kernel bit-width.

5.2.4 LSTM cells

A BiLSTM cell includes four dense layers. These layer are the cell state layer, and the input, output and forget gates. Each of these gates is a dense layer, which receives input from the previous time step and outputs a value that is used to update the cell state at the current time step. The LSTM architecture can be understood as a sequence of four interconnected dense layers. The complexity of BiLSTM is twice that of LSTM.

The computational cost of BiLSTM is

$$\begin{aligned}\mathrm{BO}_{\text{Bi-LSTM}} &= 8n_h[(n_h + n_i + 1)(b_w(b_i + b_a)) \\ &\quad +(n_h + n_i)(b_i + b_a + b_w + \log_2(n_h + n_i + 1))],\end{aligned}$$

where n_h and n_i denote the hidden units and input dimension, respectively. Moreover, b_i, b_w, and b_a represent the bit-widths of the input, weight, and activation, respectively.

5.3 Demonstration of the Quantization Gains in Experiments

In this section, we present our results on the gains obtained by quantizing the NN equalizers, for the three transmission experiments described in Chapter IV. Our main focus is to illustrate how the choice of the quantization algorithm influences the performance of these equalizers. We compare the performance before and after quantization for several PTQ and TAQ algorithms, and quantify the drop in Q-factor. Furthermore, we report significant reductions in the memory requirement and computational complexity obtained through quantization.

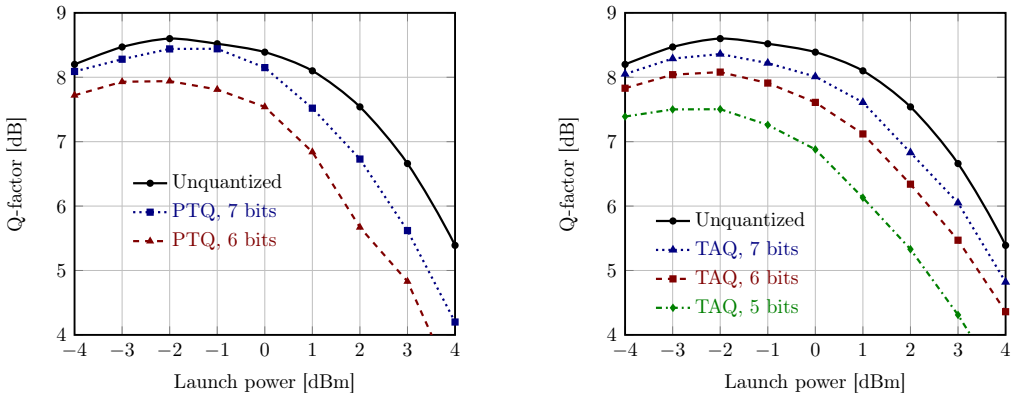


Figure 5.9: The Q-factor versus launch power in the TWC setup at several quantization rates, for (a) PTQ, (b) TAQ.

bit-width			Q-factor	
Convlutional	Dense	Quantizer	-2 dBm	2 dBm
32	32	X	8.6	7.54
6	8	uniform	8.1	6.34
6	8	ApoT	8.4	7.4

Table 5.1: Comparison of the quantization algorithms in the TWC setup.

5.3.1 TWC Experiment

We consider the TWC dual-polarization transmission experiment described in Section 4.3.1. A range of the quantization algorithms are implemented for the NN equalizer: PTQ, TAQ with STE, PTQ w/o mixed precision, and PoT quantization.

In fixed-precision PTQ, all layers are quantized at either 6 or 7 bits. In mixed-precision PTQ, 6 bits is assigned to the weights and activations of the convolutional layer, while the dense layer is given 8 bits due to its significant impact on the performance. In addition, we also explore non-uniform quantization with PoT. The TAQ technique randomly initialized the weights and activations of all layers and then quantized them at 7 and 6 bits respectively, after training with STE.

The findings of the experiment are illustrated in Fig. 5.9. Fig. 5.9(a) demonstrates that implementing PTQ at 6 bits leads to a Q-factor drop of 0.7 dB at -2 dBm, and 1.9 dB at 2 dBm. However, this technique offers a gain of 81% reduction in the memory usage and a 95% reduction in the computational complexity. As the transmission power increases, it is clear that the penalty for quantization also increases. This is because the nonlinearity of the problem intensifies, making it more challenging for the NN to manage within the

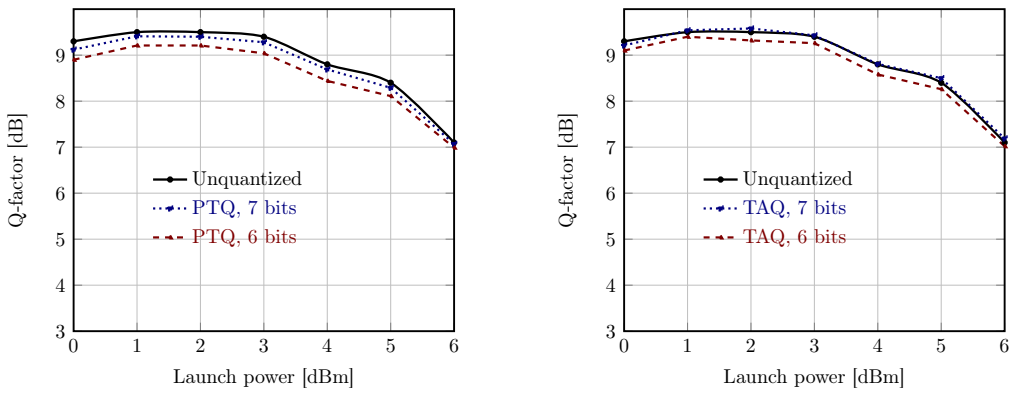


Figure 5.10: The Q-factor versus launch power in the SMF setup at several quantization rates, for (a) PTQ, (b) TAQ.

confines of weight and activation constraints.

The Q-factor improves using the TAQ as depicted in Fig. 5.9(b). The Q-factor drop is reduced to 0.5 dB at -2 dBm, and 1.2 dB at 2 dBm. Through TAQ, we observe that the NN equalizer's performance is enhanced while still achieving a reduction in the memory usage and computational complexity.

PTQ with mixed-precision surpasses the performance of TAQ at 6 bits. This is evident by the decrease in the Q-factor drop relative to the unquantized NN, which is reduced to 0.3 dB at -2 dBm and 0.34 dB at 2 dBm as presented in Tab. 5.1.

Although the convolutional layer was given 8 bits compared to 6 bits in TAQ, PTQ is still an appealing approach since quantization is performed offline after training.

It is worth noting that due to the bell-shaped distribution of the weights of the dense layer shown in Fig. 5.1, assigning more quantization symbols around the mean is a reasonable strategy. For this reason, ApoT quantization delivers the best performance, with a Q-factor penalty of less than 0.2 dB at -2 and 2 dBm. Furthermore, it has the lowest complexity since multiplications are realized using additions in ApoT quantization.

5.3.2 SMF Experiment

Next, we consider the SMF experiment described in Section 4.3.2. The quantization algorithms are similar to those in the TWC case. For the TAQ technique, the NN was initialized from the trained model to allow convergence in 20 epochs.

As depicted in Fig. 5.10 (a), implementing PTQ at 6 bits led to a Q-factor drop of 0.3 dB at 1 dBm, and 0.4 dB at 4 dBm. However, the PTQ approach resulted in a reduction

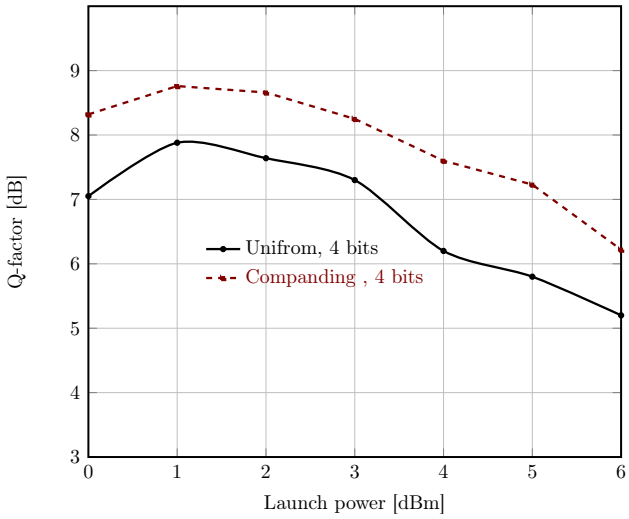


Figure 5.11: Comparison between the uniform and companding quantization of the dense layer in the SMF setup at 4 bits.

in the memory usage and computational complexity. In comparison, using TAQ, as shown in Fig. 5.10(b), the Q-factor drop was reduced to 0.1 dB at 1 dBm, and 0.2 dB at 4 dBm.

Fig. 5.11 compares the performance of the companding and uniform quantization of the dense layer in the SMF setup at 4 bits. PTQ was used to quantize the dense layer, while the other parts remained in full precision, to demonstrate the impact of the different quantizers. The results show that the companding quantization outperforms uniform quantization at low bit-widths due to the non-uniform distribution of the weights of the dense layer. APoT quantization of the dense layer at 4 bits resulted in high degradation in the performance compared to both companding and uniform quantization. Therefore, we conclude that while ApoT can provide an enhancement at large bit-widths bit regimes (as presented in the TWC setup where the NN was quantized at 8 bits), it is not a good option at low bit-widths.

5.3.3 LEAF Experiment

The LSTM neural networks can remember features in temporal sequences. This feature makes the LSTM prune to quantization noise, because small errors can be amplified by the internal activations of the LSTM. Thus, we quantize the weights and biases of the forget gate, input gate, the output gate, and the activations at the output of the LSTM. However, to limit the performance drop, the internal activations remain in full precision.

Consider the LSTM equations in described in Chapter 3. The quantizer is integrated into the internal components of the LSTM cell

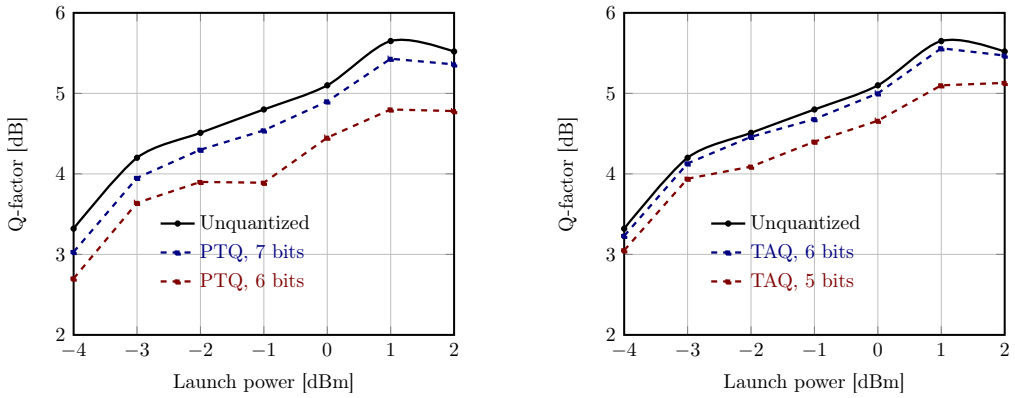


Figure 5.12: Comparison of Q-factor of the quantized BiLSTM-dense equalizer: (a) PTQ at 7 and 6 bits; (b) TAQ at 6 and 5 bits.

The operation of an LSTM unit is given by the following equations:

$$
\begin{aligned}
\mathbf{\Gamma_i}^{(t)} &= \sigma\left(Q(W_{ih})\mathbf{h}^{(t-1)} + Q(W_{ix})\mathbf{x}^{(t)} + Q(\mathbf{b_i})\right), \\
\mathbf{\Gamma_f}^{(t)} &= \sigma\left(Q(W_{fh})\mathbf{h}^{(t-1)} + Q(W_{fx})\mathbf{x}^{(t)} + Q(\mathbf{b_f})\right), \\
\mathbf{\Gamma_o}^{(t)} &= \sigma\left(Q(W_{oh})\mathbf{h}^{(t-1)} + Q(W_{ox})\mathbf{x}^{(t)} + Q(\mathbf{b_o})\right), \\
\tilde{\mathbf{c}}^{(t)} &= \tanh\left(Q(W_{ch})\mathbf{h}^{(t-1)} + Q(W_{cx})\mathbf{x}^{(t)} + Q(\mathbf{b_c})\right), \\
\mathbf{c}^{(t)} &= \mathbf{\Gamma_i}^{(t)} \odot \tilde{\mathbf{c}}^{(t)} + \mathbf{\Gamma_f}^{(t)} \odot \mathbf{c}^{(t-1)}, \\
\mathbf{h}^{(t)} &= Q\left(\mathbf{\Gamma_o}^{(t)} \odot \tanh(\mathbf{c}^{(t)})\right),
\end{aligned}
\tag{5.10}
$$

where $Q(.)$ is the quantizer function. The internal components are all quantized, except the activations in $\mathbf{\Gamma_i}^{(t)}$, $\mathbf{\Gamma_f}^{(t)}$, $\mathbf{\Gamma_o}^{(t)}$, and tanh(.) in $\tilde{\mathbf{c}}^{(t)}$. The state update equation in $\mathbf{c}^{(t)}$ is also done in floating point, and not quantized.

Fig. 5.12 (a) shows that applying PTQ at 6 bits results in a reduction of 79% in computational complexity and 81% in memory usage with a Q-factor drop of 0.9 dB and 1.2 dB observed at 1 dbm and -1 dbm, respectively. When using 5 bits, the performance noticeably decreases, resulting in a 2.7 dB drop at -1 dBm and a 2.9 dB drop at 1 dBm in Q-factor. The reason for this decline in performance is due to the intricate nature of the recurrent models that incorporate gate interactions, bi-directional dependencies, and attention, making it challenging to quantize the system at low precision without experiencing significant losses. In contrast, Fig. 5.12 (b) demonstrates that the utilization of TAQ significantly improves the system's performance. When using 6 bits, the decrease in Q-factor is much smaller, with only 0.1 dB and 0.4 dB observed at 1 dbm and -1 dbm,

respectively. Additionally, when using 5 bits, there is still a moderate Q-factor reduction of 0.3 dB at both 1 dbm and -1 dbm, but with a reduction of 82% in computational complexity and 84% in memory usage. The reason for this improvement is due to the fact that the BiLSTM has a high number of trainable parameters.

5.4 Limitations of Training Aware Quantization

In the previous section, it was demonstrated that TAQ outperforms PTQ in all simulated NN architectures and experimental setups. This is not surprising given that TAQ considers the quantization noise during the training. However, we observed that there is a minimum bit-width below which the neural network's performance with TAQ is only marginally better than that of the linear DSP. For instance, for the BiLSTM architecture, the cut-off rate is $b_c = 6$, while for the CNN-dense architecture, this value is 6 bits for the TWC setup and 5 bits for the SMF setup.

Extreme quantization at few bits has been studied in image classification. Below, we provide a few remarks on the performance of PTQ and TAQ from this literature.

- We noticed that the Q-factor is sensitive with respect to the quantization of the activations. The activation functions are nonlinear, and may amplify the quantization noise. The impact of the quantization of the activations on the Q-factor is greater than the impact of the weights.

- The impact of the quantization on the Q-factoe depends on the transmission power. As the power is increased, nonlinear distortions grow and the parameters of the NN equalizer become sensitive to small deviations.

- The back-propagation algorithm used for training relies on the reasonably accurate gradient, which is difficult to obtain with quantization (see Section 5.1).

- The NNs used in the computer vision and speech recognition are often overparameterized [5]. The over-parameterization makes the model more robust to the quantization noise, since there are more weights to mitigate the quantization noise and the error in approximating the quantizer's derivative. However, this advantage does not hold for low complexity NNs. In such cases, the quantization noise can have a significant impact on the performance of the NN.

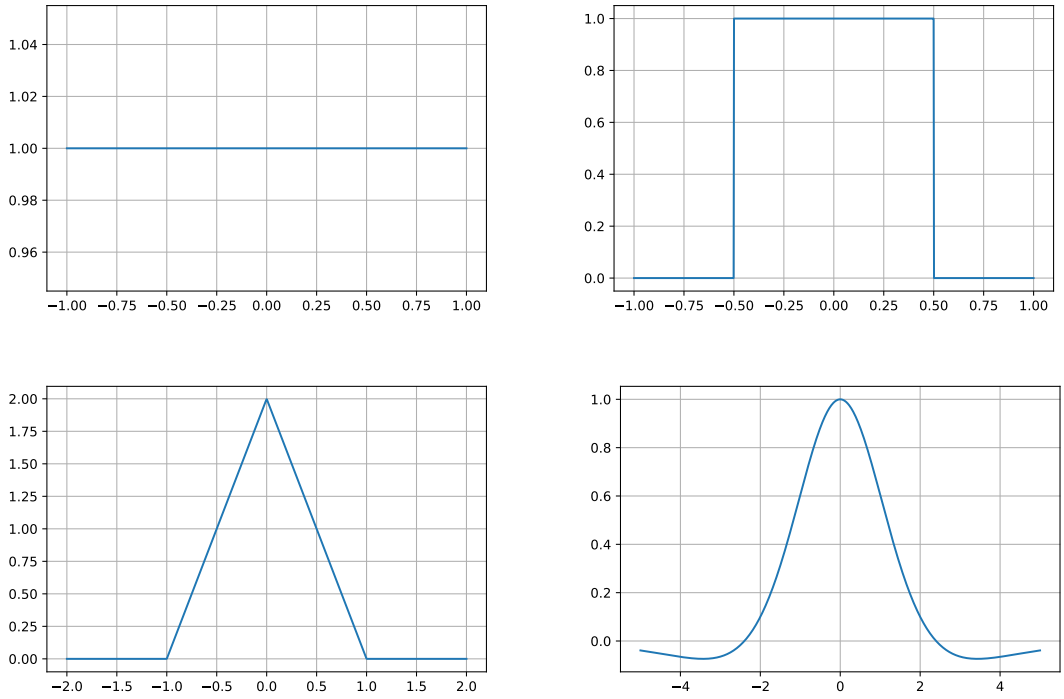


Figure 5.13: Commonly used derivatives assumed for the quantizer.

One approach to extreme quantization at low bit-widths, such as in the binary NNs, is using better approximations to the derivative than in STE [23, 25, 72, 133]. However, when we tested some of these approaches in our experiments, we did not observe any improvement in performance. Consequently, while extreme quantization using the TAQ technique has shown success in image classification, it may not be suitable for other applications, such as equalization in optical fiber.

CHAPTER 6

Quantization of Neural Network Equalizers Below 5 bits

In Chapter 5, we quantized the NN equalizers for the fiber nonlinearity mitigation in transmission experiments, using several PTQ and TAQ algorithms. These algorithms could reach 5 bits/weight and activation, while still outperforming the linear equalization. Upon extensive simulations, we concluded that the quantization below 5 bits is not useful with these algorithms, at least in their standard configurations.

This chapter is dedicated to the extreme quantization, defined as quantization up to 5 bits. We will introduce three novel algorithms for quantizing the NN equalizers: successive PTQ (SPTQ), alpha-blending (AB) and successive AB (SAB) which is a hybrid algorithm that combines the SPTQ with AB. These algorithms are iterative, incorporate ideas from PTQ and TAQ, and outperform those in Chapter 5 at a marginal cost to the complexity.

The findings of this chapter demonstrate that the weights of the NN can be quantized up to one bit, if the activations are not quantized. Further, it is shown that both weights and activations can be quantized at 2–3 bits, while still outperforming the linear equalization. Finally, we study the impact of the quantization noise arising separately from the weights and activations on the Q-factor performance of the model. This chapter is based on the journal paper [27].

6.1 Successive Post Training Quantization

This section describes the SPTQ approach for quantizing the convolutional-dense and BiLSTM-dense NNs in Chapter 4 for the fiber nonlinearity mitigation.

There are many quantization algorithms in deep learning. However, most of them have

been developed for large NNs, *e.g.,* with billions of parameters. These networks have many parameters to compensate for the quantization error. In contrast, the NNs used for fiber equalization are quite small, typically with few hundred or thousands of weights.

The SPTQ is described in [132] for general NNs, and is found to be an effective scheme for quantizing the small NNs encountered in optical communication. The main idea is to compensate for the "quantization noise" in the training. In this approach, the parameters (weights and activations) of the NN are partitioned into several sets and sequentially quantized based on a PTQ scheme from Chapter 5. In stage i, the parameters in the sets $k \leq i$ are quantized with PTQ and fixed, while those in the sets $k > i$ are trained in the full precision in order to compensate for the quantization noise resulting from the previous stages. This approach is simple and tends to perform well in practice, with a good PTQ scheme and hyper-parameter optimization [132].

The SPTQ is a combination of the PTQ and TAQ, without the complexity of TAQ, or having to address the zero gradient problem [132]. At stage i, the set of weights in the layer ℓ distinguished by an index set $\mathcal{P}_i^{(\ell)}$ is partitioned into two subsets $\mathcal{P}_{i,1}^{(\ell)}$ and $\mathcal{P}_{i,2}^{(\ell)}$ corresponding to the quantized and unquantized weights respectively, *i.e.,*

$$\mathcal{P}_i^{(\ell)} = \left\{\mathcal{P}_{i,1}^{(\ell)}, \mathcal{P}_{i,2}^{(\ell)}\right\}, \quad \mathcal{P}_{i,1}^{(\ell)} \cap \mathcal{P}_{i,2}^{(\ell)} = \emptyset. \tag{6.1}$$

The corresponding weights are denoted by $W_i^{(\ell)} \in \mathcal{P}_i^{(\ell)}$, $W_{i,1}^{(\ell)} \in \mathcal{P}_{i,1}^{(\ell)}$ and $W_{i,2}^{(\ell)} \in \mathcal{P}_{i,2}^{(\ell)}$. The model is first trained over $W_i^{(\ell)}$ in FP32. Then, the resulting weights $W_{i,1}^{(\ell)}$ are quantized under a suitable PTQ scheme. Next, $W_{i,1}^{(\ell)}$ is fixed, and the model is retrained by minimizing the loss function with respect to $W_{i,2}^{(\ell)}$, starting from the previously trained values. The second group is retrained in order to compensate the quantization noise in the first group, and make up for the loss in accuracy. In stage $i+1$, the above steps are repeated upon the substitution $\mathcal{P}_{i+1}^{(\ell)} \triangleq \mathcal{P}_{i,2}^{(\ell)}$. The weight partitioning, group-wise quantization, and retraining is repeated until the network is fully quantized.

In another version of this algorithm, the partitioning for all stages is set initially. That is to say, the weights are partitioned into a number of groups and successively quantized, such that at each stage the weights of the previous groups are quantized and fixed, and those of the remaining groups are retrained.

The hyper-parameters of the SPTQ are the choice of the quantizer function in PTQ and the partitioning scheme. There are several options for the partitioning scheme, such as random grouping, neuron grouping and local grouping. Research has demonstrated that

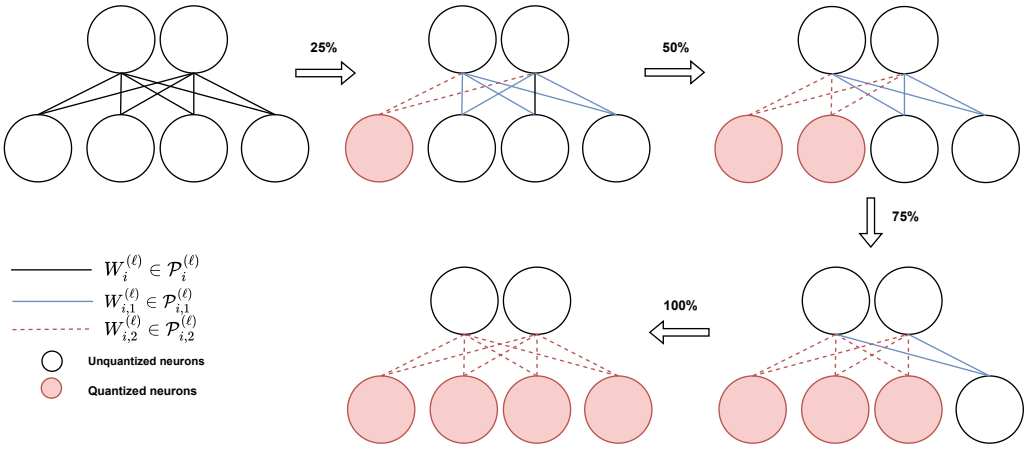


Figure 6.1: Illustration of the SPTQ. The connections with dashed red lines represent the quantized weights, while the trained weights are represented by blue lines. The NN is quantized in successive stages until all weights are quantized.

models trained with SPTQ provide classification accuracies comparable to their baseline counterparts trained and deployed in 32-bit, with fewer bits [132].

To verify the effectiveness of the SPTQ algorithm, we applied it to the convolutional-dense model in the TWC fiber transmission experiment. The experiment and hyper-parameters of the NN are explained in Chapter 5. The SPTQ is applied, by assigning a bit-width of 5 for both weights and activations of the dense layer uniformly. The convolutional layer is given 8 bits, but in our model this layer has few weights, and little impact on the complexity.

Fig. 6.2 shows the Q-factor of the SPTQ algorithm in terms of the launch power. The graph shows that even with a quantization bit-width as low as 5 bits, there is only a 0.2 dB Q-factor drop at -2 dBm, and a 0.5 dB Q-factor drop at 2 dBm. Furthermore, SPTQ generally incurs a smaller Q-factor penalty across the whole range of power, even at lower bit-widths, compared to PTQ and TAQ in Chapter 5. In comparison to our previously achieved results described in Chapter 5, the SPTQ algorithm outperforms the more complex TAQ by 2 bits at the same average signal power. However, SPTQ is marginally more complex than PTQ, since it is iterative.

The impact of the partition size on SPTQ is depicted in Fig. 6.3. By increasing the number of partitions in the dense layer, the Q-factor is enhanced. This is because a larger partition size reduces the number of the quantized weights at any given stage. A plateau in performance is observed after a certain partition size. We have observed that, as the transmission power increases, the nonlinear effects grow, making the task more challenging

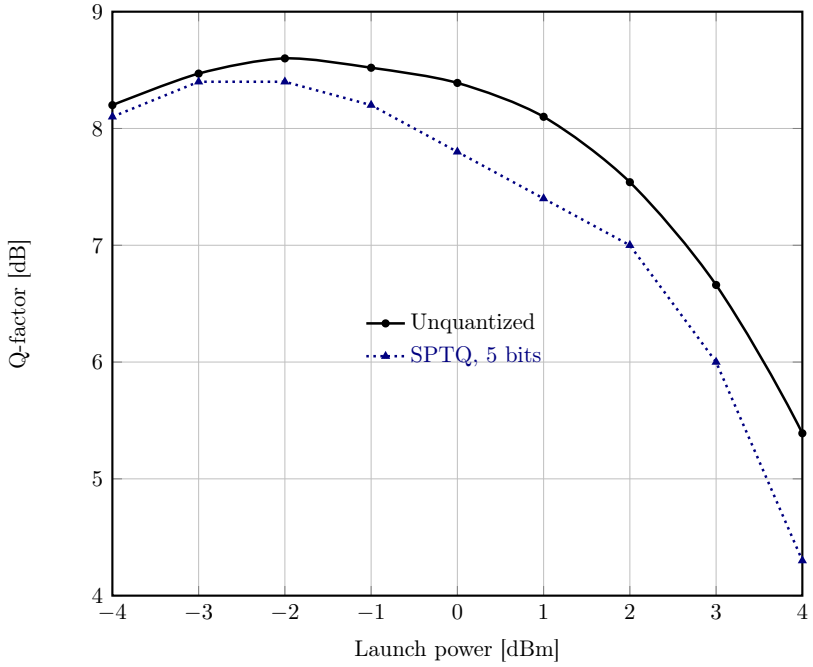


Figure 6.2: The Q-factor of SPTQ at 5 bits versus launch power.

Partition size	Q-factor							
	$\mathcal{P}_1^{(\ell)}$	$\mathcal{P}_2^{(\ell)}$	$\mathcal{P}_3^{(\ell)}$	$\mathcal{P}_4^{(\ell)}$	$\mathcal{P}_5^{(\ell)}$	$\mathcal{P}_6^{(\ell)}$	$\mathcal{P}_7^{(\ell)}$	$\mathcal{P}_8^{(\ell)}$
2	7.13	5.6	x	x	x	x	x	x
4	7.5	7.33	7.33	6.3	x	x	x	x
8	7.56	7.5	7.4	7.33	7.33	7.33	7.33	6.6

Table 6.1: The Q-factor of SPTQ at 4 bits, for different partition sizes.

for the NN, and hence, requiring more partitions to maintain a good performance.

We conducted simulations to evaluate the performance of SPTQ at 4-bit, at partitions of size 2, 4 and 8 shown in Tab. 6.1. A drop of 1.9 dB and 1.2 dB is observed compared to the unquantized NN, respectively at partitions of size 2 and 4. We noticed that the performance plateaued after using 4 partitions: the enhancement was only 0.3 dB at 8 partitions, and increasing the number of partitions did not improve the performance. Importantly, the Q-factor drop occurred in the last partition, where there were no further partitions to compensate for the quantization noise.

One way to mitigate the drop in performance in the last partition is to allocate more bits to the last partition compared to the previous partitions. This would allow for a higher level of precision for the weights and activations of the last partition, which could reduce the impact of the quantization noise in the last partition. Another approach is to use more performant quantization schemes for the last partition.

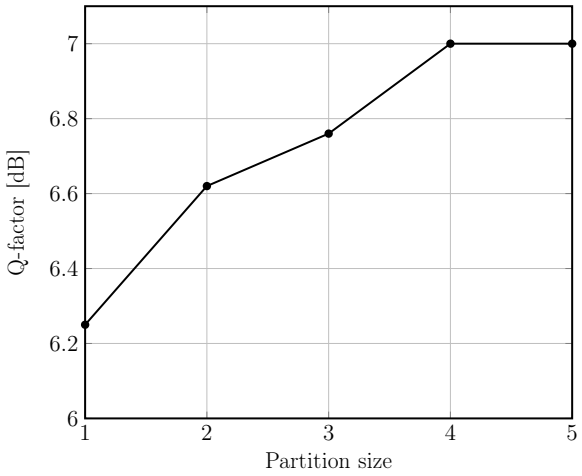


Figure 6.3: The Q-factor versus the partition size, in SPTQ at 5 bits.

6.2 Alpha-blending Quantization

In this section, we consider the application of the AB algorithm for quantizing the NN equalizers. This is a non-STE algorithm, originally proposed by [79].

Recall that TAQ faces the problem that the quantizer module introduced in the computational graph of the NN has zero gradient almost everywhere. The STE addressed this problem by assuming that this derivative is one. The AB quantization addresses the same issue by replacing the weights of the NN with a linear combination of the full precision weights and the quantized weights with a coefficient α. The loss function is therefore modified to

$$L(w, \alpha) := L((1-\alpha)w + \alpha\hat{w}).$$

The parameter α is changed from 0 to 1, as the training step i varies a training window $[T_0, T_1]$ according to:

$$\alpha = \begin{cases} 0, & i \leq T_0, \\ \left(\frac{T_1 - i}{T_1 - T_0}\right)^3, & T_0 \leq i \leq T_1, \\ 1, & i \geq T_1. \end{cases} \tag{6.2}$$

Note that, the derivative of the loss function with respect to w is

$$\begin{aligned} \frac{\partial L(w,\alpha)}{\partial w} &= L'\left((1-\alpha)w + \alpha\hat{w}\right)\left(1 - \alpha + \alpha\frac{\partial \hat{w}}{\partial w}\right) \\ &= L'\left((1-\alpha)w + \alpha\hat{w}\right)(1-\alpha), \end{aligned}$$

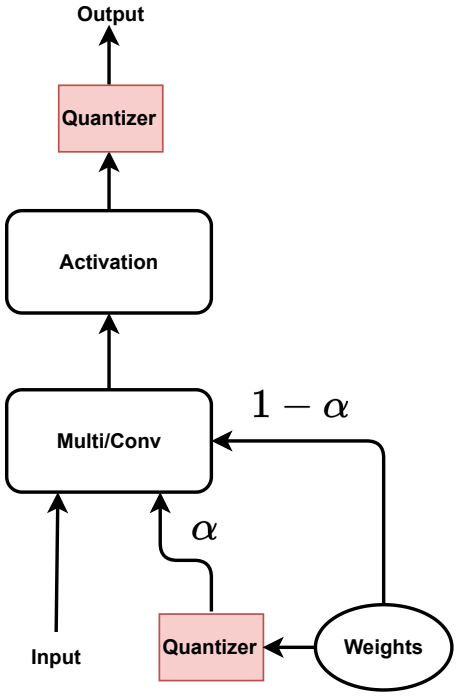


Figure 6.4: The computational graph of the NN with AB quantization during the training.The coefficient α is gradually increased from 0 to 1 during training.

where we set $\frac{\partial \hat{w}}{\partial w} = 0$. Thus, even though $\partial \hat{w}/\partial w = 0$, we have $\frac{\partial L(w,\alpha)}{\partial w} \neq 0$, and thus the weights are still updated in the gradient descent. It has been shown that the AB quantization provides an improvement over STE in different scenarios [79].

The AB quantization is integrated into the computational graph of the NN as shown in Fig. 6.4. Each weight and bias is altered during the training by taking a weighted sum of the unquantized and quantized weights. The activations are quantized with STE. This algorithm enables a smooth transition from the unquantized weights corresponding to $\alpha = 0$ to the quantized ones corresponding to $\alpha = 1$.

The AB algorithm is tested for the BiLSTM-dense equalizer receiver in the LEAF transmission experiment described in Chapter4. Fig.6.4 shows that Q-factor of the AB algorithms at 4 and 5 bits for various transmission powers. The graph demonstrates that the AB algorithm provides an enhancement over PTQ and TAQ performance presented in Chapter4. Specifically, the Q-factor drop is only 0.2 dB at -1 dBm, and 0.15 dB at 1dBm, compared to the reference unquantized NN. At 4 dBm, the Q-factor drop is 0.3 dB at -1 dBm, and 0.25 dB at 1dBm.

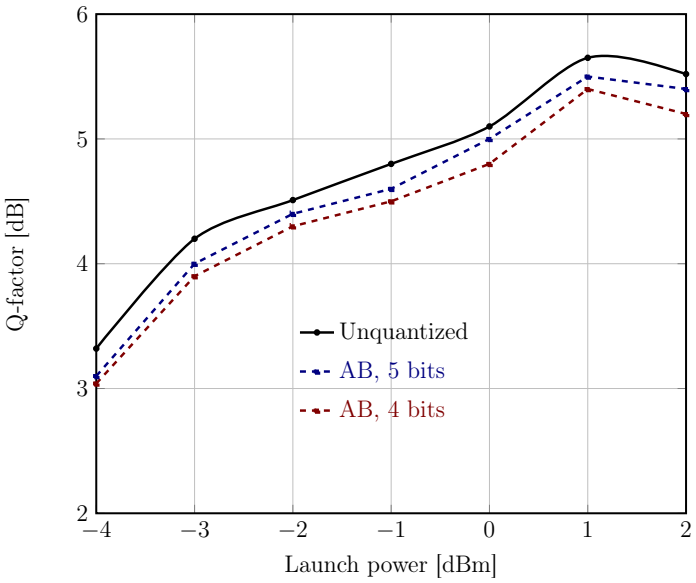


Figure 6.5: The Q-factor of AB quantization versus lunch power.

6.3 Successive Alpha-Blending Quantization

In this section, we propose SAB, an efficient performant quantization algorithm for conversion of a full-precision model to a low-precision one at 1–3 bits, depending on whether or not the activations are also quantized

SAB can be considered as a sort of combination of the SPTQ and AB quantization algorithms. It is a successive algorithm with several stages. At a given stage j, we define a partition for the weights to be quantized with two complementary sets, the same way that it was defined in SPTQ

$$\mathcal{P}_j^{(\ell)} = \left\{\mathcal{P}_{j,1}^{(\ell)}, \mathcal{P}_{j,2}^{(\ell)}\right\}, \quad \mathcal{P}_{j,1}^{(\ell)} \cap \mathcal{P}_{j,2}^{(\ell)} = \emptyset. \tag{6.3}$$

The set $\mathcal{P}_{j,1}^{(\ell)}$ corresponds to the quantized weights, and the set $\mathcal{P}_{j,2}^{(\ell)}$ to the unquantized weights. First, the weights of the quantized set $\mathcal{P}_{j,1}^{(\ell)}$ are updated according to AB scheme defined in 6.2

$$W_{j,1}^{(\ell)} = (1-\alpha)W_{j,1}^{(\ell)} + \alpha\hat{W}_{j,1}^{(\ell)}, \tag{6.4}$$

where α is the value in the sequence (6.2) at $i = T_0$. Then, the weights $W_{j,1}^{(\ell)}$ are kept fixed while the values of $W_{j,2}^{(\ell)}$ are retrained from their previously values. Next, α is incremented to the value in the sequence (6.2) at $i = T_0 + 1$. The above process is repeated until $\alpha = 1$ is reached at $i = T_1$, where all weights in $W_{j,1}^{(\ell)}$ are fully quantized. The algorithm

Quantization scheme	bit-width	Q-factor
Unquantized	32	7.5
SPTQ	4	6.3
AB	4	6.3
SAB	4	7.0

Table 6.2: Comparison of the quantization algorithms, in the TWC experiment. The SPTQ and SAB schemes have a partition of size 4.

then advances to the next stage $j+1$, by partitioning $\mathcal{P}_{j,2}^{(\ell)}$ into two complementary sets. The last partition is trained with the AB algorithm instead of being fixed, to address the problem of the performance drop in the last set that was encountered in SPTQ scheme.

It is important to note that SAB is not exactly a hybrid of SPTQ and AB: the successive retraining strategy is distributed in the AB algorithm with respect to α. Therefore, SAB quantization improves upon SPTQ and AB quantization, since each partition is not quantized in one shot, rather is incrementally quantized by increasing α. This allows the trained set $\mathcal{P}_{j,2}^{(\ell)}$ to adapt to the changes in $\mathcal{P}_{j,1}^{(\ell)}$. Instead of fixing the last partition as in the SPTQ scheme, the AB algorithm is applied to train the last partition and fix the quantization noise. This modification leads to a reduction in the drop in performance occurred in the last partition. By allowing the weights in the last partition to be adjusted using the AB algorithm, the quantization noise is better compensated for.

To assess the effectiveness of our proposed quantization scheme, we applied it to the convolutional-dense equalizer in the both TWC and SMF experiments.

6.3.1 TWC Experiment

In the TWC setup, in a first study, we considered a partition of size 4 with the weights and activations in each partition set quantized at 4 bits. We compare the Q-factor performance of the SAB, SPTQ and AB quantization schemes in Tab. 6.2. The results indicate that SAB outperforms the other two methods, with a performance drop of only 0.5 dB compared to the original NN. In contrast, SPTQ and AB resulted in a 1.2 dB drop in performance.

The performance can be increased by applying mixed precision on the partitions sets. Giving more bits to last partition reduces the Q-factor drop. Thus, in a second study, we considered a partition of size 4 with the weights and activations in the first 3 partition sets quantized at 4 bits, and in the last partition set at 6 bits. The results are shown in Fig. 6.6(a). The graph indicates a Q-factor drop of 0.17 dB at -2 dBm and 0.24 dB at 2 dBm, along with an 86% reduction in memory usage and a 94% reduction in computational

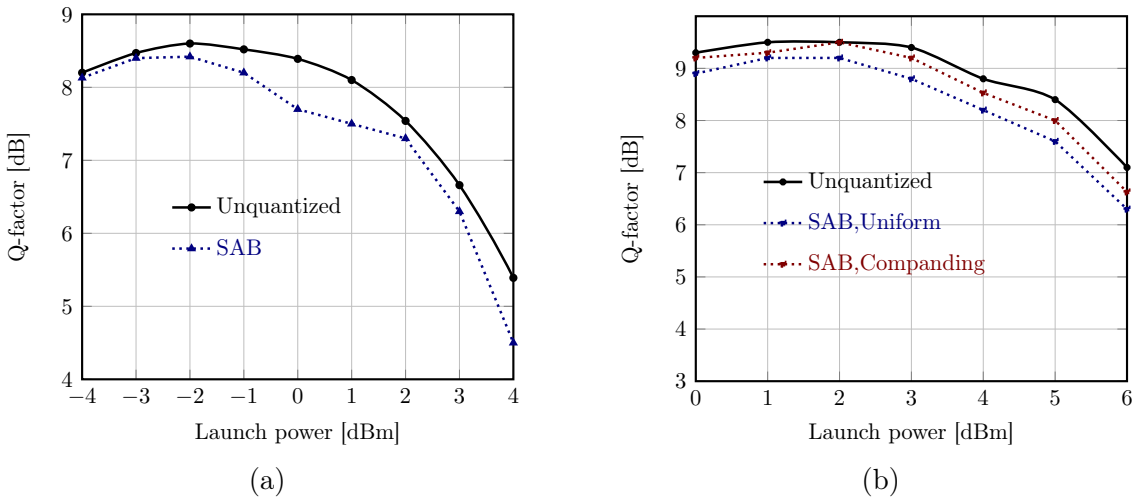


Figure 6.6: The Q-factor of SAB quantization versus launch power, for the convolutional-dense equalizer, in a) TWC experiment, and (b) SMF experiment.

complexity.

6.3.2 SMF Experiment

For the SMF setup, we considered a partition of size 4 with the weights and activations in the first 3 partition sets quantized at 3 bits, and in the last partition set at 6 bits. The activations for all partition sets were quantized at 3 bits. The uniform quantization, and non-uniform quantization using the μ companding, were applied separately.

Fig. 6.6(b) shows the performance results for the application of SAB quantization to the SMF experiment. Uniform quantization results in a Q-factor drop of 0.3 dB at 1 dBm, and 0.6 dB at 4 dBm. This approach also demonstrated a significant reduction in memory usage and computational complexity, by 88% and 94%, respectively. However, by applying the companding quantization, the Q-factor drop was reduced to 0.2 dB at 1 dBm.

6.4 Quantization of Weights, but not Activations

In this section, we focus on quantizing the weights of the NN by allowing the activations have 8 bits. This is because the drop in performance is primarily due to the quantization of the activations. By quantizing the weights at low resolutions but not activation, we obtain a significant reduction in memory and storage, at the cost of increase in computational complexity. The increase in complexity is because in the MAC operation, the bit-width of the output increases with additions and multiplication, and the activations are responsible

Bit width					Q-factor
$\mathcal{P}_1^{(\ell)}$	$\mathcal{P}_2^{(\ell)}$	$\mathcal{P}_3^{(\ell)}$	$\mathcal{P}_4^{(\ell)}$	Activation	
32	32	32	32	32	9.5
3	3	3	3	8	9.2
2	2	2	2	8	8.0
1	1	1	4	8	8.9

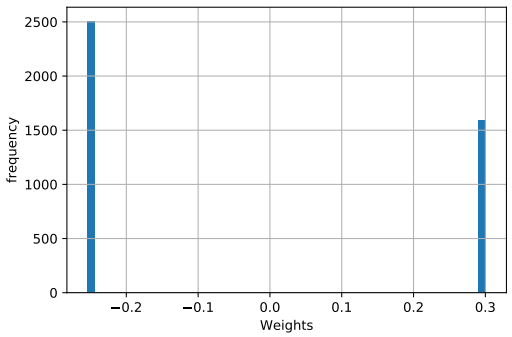
Table 6.3: Q-factor performance of SAB scheme with 8-bit quantized activations on convolutional-dense receiver in SMF transmission setup at optimal power.

for reducing the width. We note that, we still quantize the activations, but not below 8 bits (because reducing the resolution from 32 to 8 bits has little impact).

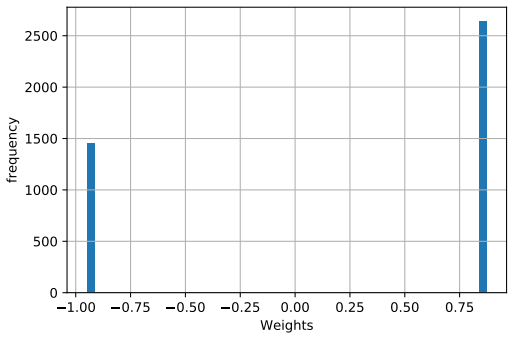
We first present our results in a study where the weights of the convolutional-dense receiver in the SMF experiment are quantized using the SAB quantization with fixed-precision, while the activations operate at 8 bits. The results are presented in Tab. 6.3, which shows that the Q-factor drop is minimal, with the dense layer quantized at as low as 3 bits. This is a significant improvement over the previous results where the last partition was given 6 bits due to the loss in performance caused by quantizing the activations, as shown in Fig. 6.6. However, it should be noted that even relaxing the condition on the quantization of the activations, the dense layer cannot perform below 2 bits without significant degradation in performance. It can be seen in Tab. 6.3 with 2 bits, the drop in performance is 1.5 dB compared to the original model. This is attributed to the influence of the last partition, which handles the quantization noise for all partitions. Therefore, the last partition remains a major limiting factor in achieving higher compression rates without sacrificing performance.

Finally, we present our results on the binary NNs, improving upon the previous study with mixed-precision. As before, we let activations operate at 8 bits. We partition the weights into 4 sets. The first three partitions are quantized at 1 bit, and the last one at 4 bits. We observe just a moderate degradation in Q-factor: 0.6 dB. This result is important, because it demonstrates for the first time that low-complexity binary NNs can mitigate nonlinearities in optical fiber communication.

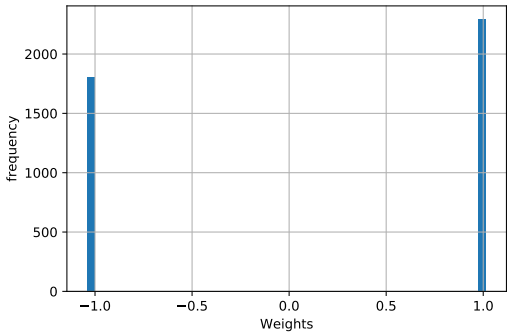
Binarization usually takes advantage of the fact that the NNs used in deep learning often have a large number of trainable parameters. For example, the ResNet from 2016 [53] has 26.6 million, however, in our case, the NN has only 16,000 parameters.



(a) $\mathcal{P}_1^{(\ell)}$



(b) $\mathcal{P}_2^{(\ell)}$



(c) $\mathcal{P}_3^{(\ell)}$

Figure 6.7: The distribution of the weights of the first three sets in the partition.

CHAPTER 7

Conclusions

This dissertation is dedicated to low-complexity nonlinearity mitigation in dual polarization optical fiber transmission experiments using quantized NNs. NNs are data-driven, do not require the knowledge of the channel, and can adapt to the changes in the transmission medium to compensate the distortions more efficiently.

Chapter 2 and 3 provide the background and review material required to understand the research. We reviewed the principles of the digital transmission over optical fiber. We explained the linear and nonlinear effects in dual-polarization transmission over optical fibers, and described the coherent receiver with the DSP chain, for mitigating the chromatic dispersion, PMD and carrier phase offset. We introduced the multi-layer perceptron, convolutional and long short-term memory NNs, and explained the training of the NNs using the gradient descent and the backpropagation algorithm.

The contributions of this work are briefly outlined below.

7.1 Two NNs for Nonlinearity Mitigation in Transmission Experiments

We quantified the gains that can be achieved by mitigating the fiber nonlinearities with DBP. The problem with DBP is that, it can be computationally complex, motivating research in alternative equalizers such as NNs.

We demonstrated the potential of using NNs for mitigating the fiber nonlinearity in three dual-polarization transmission experiments: a 9x50km TWC, a 9x110km SMF, and a 17x70km LEAF setup. We proposed two low-complexity NN-based nonlinear equalizers, a convolutional-dense and a BiLSTM-dense model, placed at the end of the linear DSP

for mitigating the nonlinearities. The findings indicated that the Q-factor of the NNs and DBP are comparable, both of which greater than the performance of the linear DSP, particularly in the nonlinear regime.

Quantization is one approach to reduce the size of the NN, which is particularly important in the low-power chips which have limited memory and computational resources. We explored quantizing the proposed models with PTQ and TAQ, highlighting how these quantization schemes can be applied to the internal structures of the LSTM cell.

7.2 Quantization Above 5 Bits

In this case, the findings showed that TAQ with STE outperforms PTQ, since it mitigates the quantization noise to some extent. The Q-factor of the quantized NN increases with the number of neurons. This is probably because the NN can better compensate for the quantization noise with a large number of trainable parameters. The Q-factor drop due to quantization increases with the transmission power or distance, which makes sense, since the BER is high at high powers. It was shown that the BiLSTM based receiver is particularly prone to the quantization noise in PTQ, since the error is amplified by the internal activations of the LSTM cell. We explored mixed-precision quantization, and determined the impact of the number of bits in each layer on the loss function.

In the convolutional-dense receiver, the dense layer has a greater impact on the performance compared to the convolutional layer, mainly due to containing more trainable parameters. Quantizing this layer with more bits lowered the BER notably. An improvement over uniform quantization can be achieved through non-uniform quantization when the weight distribution of the layer is non-uniformly distributed. APoT is a nonuniform quantization scheme where the quantized values have a power-of-two representation. This technique is suitable for hardware implementation, since the multiplications are converted to additions. We noted that APoT provides a similar or slightly better performance than the uniform quantization at high bit widths. At low bit widths, uniform quantization is better than APoT.

Companding quantization is a non-uniform quantization scheme that can avoid the complex hardware requirements needed to accurately represent the non-uniform quantization symbols by performing uniform quantization on a nonlinearly-transformed signal. Our results demonstrate that implementing this scheme for the dense layer can provide a very good performance, especially at low bit widths, as shown in Fig. 5.11 and 6.6.

It was demonstrated that TAQ outperforms PTQ in all simulated setups. However, it was observed that there is a minimum bit-width that TAQ works reasonably well. The limitations of TAQ were explained in Section 5.4. Since the quantization function is piece-wise flat, it has a derivative that is almost zero everywhere, which does not work well with the back-propagation algorithm. Therefore, an approximation of the derivative using, *e.g.,* the STE method, is necessary. The quantization of the activations has a greater impact on performance since it directly affects the output of the layer. Also, the proposed low complexity models are not overparameterized, and therefore cannot handle extreme low bit quantization.

One approach that we tried to overcome the poor performance of TAQ in low bit-widths was to use better derivative approximations. We tried various derivatives, however, they did not lead to any significant improvement in performance. However, not all types of derivatives have been tested in this work, and further research is needed to draw a robust conclusion.

7.3 Quantization Below 5 Bits

Below 5 bits, one quantization technique that avoids the need for derivative approximation is SPTQ. This approach involves partitioning the trainable parameters of the NN into two distinct sets: the first set is quantized with PTQ and fixed, and then the second set is trained with full precision to mitigate the noise caused by quantizing the first set. This process is iterated until all weights are quantized. This approach compensates for the quantization noise during training.

Compared to the PTQ and TAQ schemes presented in Chapter 5, the SPTQ algorithm achieved better results than the more complex TAQ algorithm by 2 bits in the same experiment.

As discussed in Section 6.1, the performance of SPTQ is mainly limited by the quantization of the weights of the last partition. Our results indicate that the drop in performance occurs primarily during the quantization of this partition. This is likely due to the fact that there are no remaining partitions to help compensate for the quantization error. Therefore, it is important to explore alternative quantization techniques that can address the drop in the performance of the last partition.

A second approach that we tried to address the problem of the zero derivative of the quantizer is the AB quantization. This method modifies the weights to be a linear

combination of the quantized and unquantized weights, as shown in Fig. 6.4. Consequently, even though the derivative of the quantizer with respect to its input is zero, the gradient of the loss function with respect of weights does not vanish.

The AB technique was applied to the BiLSTM-dense model, as illustrated in Fig. 6.5. It was found that this approach enabled the quantization of the weights and activations at 4 bits, with only a slight decrease in performance compared to the TAQ.

In Section 6.3 we presented our proposed quantization scheme SAB. This approach can be viewed as sort of a hybrid of the SPTQ and AB quantization, and improves upon the SPTQ and AB quantization. The main advantage of SAB is that it allows each partition to be quantized incrementally, making it easier for the trained set to adapt to the small changes in the quantized set. In contrast to the SPTQ, the AB algorithm is used to train the last partition and fix the quantization noise. This modification leads to a reduction in the drop in performance in the last partition. By allowing the weights in the last partition to be adjusted using the AB algorithm, the quantization noise is effectively compensated for, resulting in improved performance compared to both SPTQ and AB quantization.

The SAB approach can be implemented with mixed-precision quantization, assigning a higher bit width to the last partition set, and lower bit widths to the initial partition sets. This can further enhance the performance of the SAB quantization, by gradually increasing the bit widths of the partition sets to better match their sensitivity to the quantization noise.

The results of Chapter VI demonstrated that SAB quantization improved upon all the previously presented quantization methods. This was particularly evident in the convolutional-dense receiver, where our scheme allowed for quantization of weights and activations with as low as 3 bits while maintaining good performance. If the activations are not quantized, SAB quantization of the weights at 1–2 bits still notably outperforms linear equalization.

Bibliography

[1] Govind P Agrawal. "Nonlinear fiber optics". In: *Nonlinear Science at the Dawn of the 21st Century*. Springer, 2000, pp. 195–211.

[2] Govind P. Agrawal. "Chapter 2 - Pulse propagation in fibers". In: *Nonlinear Fiber Optics (Sixth Edition)*. Ed. by Govind P. Agrawal. Sixth Edition. Academic Press, 2019, pp. 27–55. ISBN: 978-0-12-817042-7.

[3] Govind P. Agrawal. "Optical communication: its history and recent progress". In: *Optics in our time* (2016), pp. 177–199.

[4] Nikola Alić et al. "Signal statistics and maximum likelihood sequence estimation in intensity modulated fiber optic links containing a single optical preamplifier". In: *Optics Express* 13.12 (2005), pp. 4568–4579.

[5] Zeyuan Allen-Zhu, Yuanzhi Li, and Yingyu Liang. "Learning and Generalization in Overparameterized Neural Networks, Going Beyond Two Layers". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[6] Abdelkerim Amari, Philippe Ciblat, and Yves Jaouën. "Fifth-order Volterra series based nonlinear equalizer for long-haul high data rate optical fiber communications". In: *2014 48th Asilomar Conference on Signals, Systems and Computers*. IEEE. 2014, pp. 1367–1371.

[7] Ron Banner, Yury Nahshan, and Daniel Soudry. "Post training 4-bit quantization of convolutional networks for rapid-deployment". In: *Proc . Adv . Neural Inf . Process . Sys*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019.

[8] Ron Banner et al. "Scalable methods for 8-bit training of neural networks". In: *Proc . Adv . Neural Inf . Process . Sys* 31 (2018).

[9] Philippe M Becker, Anders A Olsson, and Jay R Simpson. *Erbium-doped fiber amplifiers: fundamentals and technology*. Elsevier, 1999.

[10] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. "Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation". In: *arXiv:1308.3432* (Aug. 2013), pp. 1–12.

[11] R Berry, D Brace, and I Ravenscroft. "Optical fiber system trials at 8 Mbits/s and 140 Mbits/s". In: *IEEE Transactions on Communications* 26.7 (1978), pp. 1020–1027.

[12] Davis Blalock et al. "What is the state of neural network pruning?" In: *Proceedings of machine learning and systems* 2 (2020), pp. 129–146.

[13] Craig F Bohren and Donald R Huffman. *Absorption and scattering of light by small particles*. John Wiley & Sons, 2008.

[14] R. M. Bütler et al. "Model-Based Machine Learning for Joint Digital Backpropagation and PMD Compensation". In: *Journal of Lightwave Technology* 39.4 (2021), pp. 949–959.

[15] Clara Catanese et al. "A Fully Connected Neural Network to Mitigate 200G DP-16-QAM Transmission System Impairments". In: *OSA Advanced Photonics Congress (AP) 2020 (IPR, NP, NOMA, Networks, PVLED, PSC, SPPCom, SOF)*. Optical Society of America, 2020, SpTh3I.1.

[16] Grigorios Charalabopoulos, Peter Stavroulakis, and A Hamid Aghvami. "A frequency-domain neural network equalizer for OFDM". In: *GLOBECOM'03. IEEE Global Telecommunications Conference (IEEE Cat. No. 03CH37489)*. Vol. 2. IEEE. 2003, pp. 571–575.

[17] Zhaomin Chen et al. "Autoencoder-based network anomaly detection". In: *2018 Wireless Telecommun. Symp.* IEEE. 2018, pp. 1–5.

[18] Yoni Choukroun et al. "Low-bit Quantization of Neural Networks for Efficient Inference". In: *Proc. IEEE Int. Conf. Comput. Vis. (ICCV),* Oct. 2019, pp. 3009–3018.

[19] Andrew R Chraplyvy, Robert W Tkach, and Kenneth L Walker. *Optical fiber for wavelength division multiplexing*. US Patent 5,327,516. July 1994.

[20] AR Chraplyvy. "The coming capacity crunch,[in Proc". In: *ECOC, Vienna, Austria* (2009).

[21] AR Chraplyvy et al. "8* 10 Gb/s transmission through 280 km of dispersion-managed fiber". In: *IEEE photonics technology letters* 5.10 (1993), pp. 1233–1235.

[22] Cisco Systems. *Cisco Annual Internet Report (2018–2023)*. White Paper. 2018.

[23] Matthieu Courbariaux et al. *Binarized Neural Networks: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1*. 2016. arXiv: 1602.02830.

[24] Cristian B Czegledi et al. "Polarization-mode dispersion aware digital backpropagation". In: *ECOC 2016; 42nd European Conference on Optical Communication*. VDE. 2016, pp. 1–3.

[25] Sajad Darabi et al. *Regularized Binary Network Training*. 2020. arXiv: 1812.11800.

[26] Jamal Darweesh et al. "Quantization of Neural Networks for Nonlinearity Mitigation in an Optical Fiber Transmission Experiment". In: *European Conf. Opt. Commun.* Sept. 2022.

[27] Jamal Darweesh et al. "Quantization of Neural Networks for Nonlinearity Mitigation in Optical Fiber Transmission Experiments". In: (Sept. 2023). To be submitted.

[28] Jamal Darweesh et al. "Successive Quantization of the Neural Network Equalizers in Optical Fiber Communication". In: *Opto-Electronics and Communications Conference*. July 2023.

[29] Stavros Deligiannidis et al. "Compensation of Fiber Nonlinearities in Digital Coherent Systems Leveraging Long Short-Term Memory Neural Networks". In: *J. Lightwave Technol.* 38.21 (Nov. 2020), pp. 5991–5999.

[30] Stavros Deligiannidis et al. "Compensation of fiber nonlinearities in digital coherent systems leveraging long short-term memory neural networks". In: *IEEE J. Lightw. Technol.* 38.21 (2020), pp. 5991–5999.

[31] F Derr. "Optical QPSK transmission system with novel digital receiver concept". In: *Electronics Letters* 23.27 (1991), pp. 2177–2179.

[32] E. Desurvire, J. R. Simpson, and P. C. Becker. "High-gain erbium-doped traveling-wave fiber amplifier". In: *Opt. Lett.* 12.11 (Nov. 1987), pp. 888–890.

[33] Zhen Dong et al. "HAWQ-V2: Hessian Aware trace-Weighted Quantization of Neural Networks". In: *Proc . Adv . Neural Inf . Process . Sys.* Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 18518–18529.

[34] Zhen Dong et al. “HAWQ: Hessian AWare Quantization of Neural Networks With Mixed-Precision”. In: *Proc. IEEE Int. Conf. Comput. Vis. (ICCV),* Oct. 2019.

[35] Liang B. Du and Arthur J. Lowery. “Improved single channel backpropagation for intra-channel fiber nonlinearity compensation in long-haul optical communication systems”. In: *OSA* 18 (July 2010), pp. 17075–17088.

[36] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. “Neural architecture search: A survey”. In: *The Journal of Machine Learning Research* 20.1 (2019), pp. 1997–2017.

[37] René-Jean Essiambre et al. “Capacity Limits of Optical Fiber Networks”. In: *Journal of Lightwave Technology* 28.4 (2010), pp. 662–701.

[38] A Farbert. “Performance of a 10.7 Gb/s receiver with digital equaliser using maximum likelihood sequence estimation”. In: *European Conference and Exhibition on Optical Communication (ECOC), Sep. 2004*. 2004.

[39] F Forghieri, RW Tkach, and AR Chraplyvy. “Fiber nonlinearities and their impact on transmission systems”. In: *Optical Fiber Telecommunications IIIA* 1 (1997).

[40] P. J. Freire et al. “Complex-Valued Neural Network Design for Mitigation of Signal Distortions in Optical Links”. In: *Journal of Lightwave Technology* (2020), pp. 1–1.

[41] Pedro J Freire et al. “Reducing Computational Complexity of Neural Networks in Optical Channel Equalization: From Concepts to Implementation”. In: *IEEE J. Lightw. Technol.* (2023).

[42] Pedro J. Freire et al. “Performance Versus Complexity Study of Neural Network Equalizers in Coherent Optical Systems”. In: *IEEE J. Lightw. Technol.* 39.19 (Oct. 2021), pp. 6085–6096.

[43] Jiabao Gao et al. “Online deep neural network for optimization in wireless communications”. In: *IEEE Wireless Communications Letters* 11.5 (2022), pp. 933–937.

[44] S. J. Garth and C. Pask. “Four-photon mixing and dispersion in single-mode fibers”. In: *Opt. Lett.* 11.6 (June 1986), pp. 380–382.

[45] Amir Gholami, Michael W Mahoney, and Kurt Keutzer. “An integrated approach to neural network design, training, and inference”. In: *Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep* (2020).

[46] Boris Ginsburg et al. *Tensor processing using low precision format*. US Patent App. 15/624,577. Dec. 2017.

[47] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

[48] J. P. Gordon and L. F. Mollenauer. "Phase noise in photonic communications systems using linear amplifiers". In: *Opt. Lett.* 15.23 (Dec. 1990), pp. 1351–1353.

[49] Yunhui Guo. "A survey on methods and theories of quantized neural networks". In: *arXiv preprint arXiv:1808.04752* (2018).

[50] Suyog Gupta et al. "Deep learning with limited numerical precision". In: *International conference on machine learning*. PMLR. 2015, pp. 1737–1746.

[51] C. Häger and H. D. Pfister. "Nonlinear Interference Mitigation via Deep Neural Networks". In: *2018 Optical Fiber Communications Conference and Exposition (OFC)*. 2018, pp. 1–3.

[52] Song Han et al. "Learning both weights and connections for efficient neural network". In: *NeurIPS* 28 (2015).

[53] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[54] Pinjing He et al. "A fiber nonlinearity compensation scheme with complex-valued dimension-reduced neural network". In: *IEEE Photon. J* 13.6 (2021), pp. 1–7.

[55] Jeff Hecht. "City of Light: The Story of Fiber Optics". In: 1999.

[56] Andrew G Howard et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications". In: *arXiv preprint arXiv:1704.04861* (2017).

[57] Gao Huang et al. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[58] Itay Hubara et al. "Improving post training neural quantization: Layer-wise calibration and integer programming". In: *arXiv preprint arXiv:2006.10518* (2020).

[59] *Infrapedia*. https://www.infrapedia.com. Accessed on May 18, 2023.

[60] Yani Ioannou et al. "Deep Roots: Improving CNN Efficiency With Hierarchical Filter Groups". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[61] E. Ip and J. M. Kahn. “Compensation of Dispersion and Nonlinear Impairments Using Digital Backpropagation”. In: 26.20 (Oct. 2008), pp. 3416–3425.

[62] Benoit Jacob et al. “Quantization and training of neural networks for efficient integer-arithmetic-only inference”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 2704–2713.

[63] Mutsam A Jarajreh et al. “Artificial neural network nonlinear equalizer for coherent optical OFDM”. In: *IEEE Photonics Technology Letters* 27.4 (2014), pp. 387–390.

[64] Yang Ji et al. “An electronic mach–zehnder interferometer”. In: *Nature* 422.6930 (2003), pp. 415–418.

[65] Boris Karanov et al. “End-to-End Deep Learning of Optical Fiber Communications”. In: *Journal of Lightwave Technology* 36.20 (2018), pp. 4843–4855.

[66] Tetsuya Kawanishi. “Parallel Mach-Zehnder modulators for quadrature amplitude modulation”. In: *IEE. Electron Express.* 8.20 (2011), pp. 1678–1688.

[67] R. S. Kerdock and D. H. Wolaver. “Atlanta fiber system experiment: results of the atlanta experiment”. In: *The Bell System Technical Journal* 57.6 (1978), pp. 1857–1879.

[68] Toshiaki Koike-Akino et al. “Zero-Multiplier Sparse DNN Equalization for Fiber-Optic QAM Systems with Probabilistic Amplitude Shaping”. In: *2021 European Conference on Optical Communication (ECOC)*. 2021, pp. 1–4.

[69] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “Imagenet classification with deep convolutional neural networks”. In: *Commun. ACM* 60.6 (2017), pp. 84–90.

[70] E-H Lee, KH Kim, and HK Lee. “Nonlinear effects in optical fiber: Advantages and disadvantages for high capacity all-optical communication application”. In: *Optical and quantum electronics* 34 (2002), pp. 1167–1174.

[71] Andreas Leven et al. “Frequency Estimation in Intradyne Reception”. In: *IEEE Photonics Technology Letters* 19.6 (2007), pp. 366–368.

[72] Fengfu Li et al. *Ternary Weight Networks*. 2022. arXiv: 1605.04711.

[73] Xiaoxu Li et al. “Electronic post-compensation of WDM transmission impairments using coherent detection and digital signal processing”. In: *Optics Express* 16.2 (2008), pp. 880–888.

[74] Yuhang Li, Xin Dong, and Wei Wang. “Additive powers-of-two quantization: An efficient non-uniform discretization for neural networks”. In: *arXiv:1909.13144* (2019).

[75] Darryl Lin, Sachin Talathi, and Sreekanth Annapureddy. “Fixed Point Quantization of Deep Convolutional Networks”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, June 2016, pp. 2849–2858.

[76] Richard A Linke and Alan H Gnauck. “High-capacity coherent lightwave systems”. In: *Journal of Lightwave Technology* 6.11 (1988), pp. 1750–1769.

[77] Ling Liu et al. “Intrachannel nonlinearity compensation by inverse Volterra series transfer function”. In: *Journal of Lightwave Technology* 30.3 (2011), pp. 310–316.

[78] Zechun Liu et al. “Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm”. In: *Euro Conf. Comp. Vision*. 2018, pp. 722–737.

[79] Zhi-Gang Liu and Matthew Mattina. “Learning low-precision neural networks without straight-through estimator (ste)”. In: *arXiv preprint arXiv:1903.01061* (2019).

[80] Doug McGhan et al. “5120-km RZ-DPSK transmission over G. 652 fiber at 10 Gb/s without optical dispersion compensation”. In: *IEEE Photonics Technology Letters* 18.2 (2006), pp. 400–402.

[81] Robert J Mears et al. “Low-noise erbium-doped fibre amplifier operating at 1.54 μm”. In: *Electronics Letters* 19.23 (1987), pp. 1026–1028.

[82] Szymon Migacz. “8-bit inference with tensorrt”. In: *GPU technology conference*. Vol. 2. 4. 2017, p. 5.

[83] P Minzioni et al. “Study of the Gordon–Mollenauer effect and of the optical-phase-conjugation compensation method in phase-modulated optical communication systems”. In: *IEEE Photonics Journal* 2.3 (2010), pp. 284–291.

[84] Paolo Minzioni and Alessandro Schiffini. “Unifying theory of compensation techniques for intrachannel nonlinear effects”. In: *Optics express* 13.21 (2005), pp. 8460–8468.

[85] Paolo Minzioni et al. “Experimental demonstration of nonlinearity and dispersion compensation in an embedded link by optical phase conjugation”. In: *IEEE photonics technology letters* 18.9 (2006), pp. 995–997.

[86] Partha P Mitra and Jason B Stark. “Nonlinear limits to the information capacity of optical fibre communications”. In: *Nature* 411.6841 (2001), pp. 1027–1030.

[87] L.F. Mollenauer, S.G. Evangelides, and H.A. Haus. “Long-distance soliton propagation using lumped amplifiers and dispersion shifted fiber”. In: *Journal of Lightwave Technology* 9.2 (1991), pp. 194–197.

[88] Agostino Moncalvo and Federico Tosco. “European field trials and early applications in telephony”. In: *IEEE Journal on Selected Areas in Communications* 1.3 (1983), pp. 398–403.

[89] Markus Nagel et al. “Up or Down? Adaptive Rounding for Post-Training Quantization”. In: *in Proc. 37th Int. Conf. Mach. Learn.* Ed. by Hal Daumé III and Aarti Singh. Vol. 119. PMLR, 13–18 Jul 2020, pp. 7197–7206.

[90] Ali Bou Nassif et al. “Speech recognition using deep neural networks: A systematic review”. In: *IEEE access* 7 (2019), pp. 19143–19165.

[91] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. “In search of the real inductive bias: On the role of implicit regularization in deep learning”. In: *arXiv preprint arXiv:1412.6614* (2014).

[92] David M Pepper and Amnon Yariv. “Compensation for phase distortions in nonlinear media by phase conjugation”. In: *Optics letters* 5.2 (1980), pp. 59–60.

[93] Zoran Peric et al. “Robust 2-bit quantization of weights in neural network modeled by Laplacian distribution”. In: *Adv. Electr. Comput. Eng* 21 (2021), pp. 3–10.

[94] S. Perrin. “Deployment and Service Activation at 100G and Beyond”. In: (2015).

[95] Zhongnan Qu et al. “Adaptive loss-aware quantization for multi-bit networks”. In: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 2020, pp. 7988–7997.

[96] Sujan Rajbhandari, Zabih Ghassemlooy, and Maia Angelova. “Effective denoising and adaptive equalization of indoor optical wireless channel with artificial light using the discrete wavelet transform and artificial neural network”. In: *Journal of Lightwave technology* 27.20 (2009), pp. 4493–4500.

[97] Diego Argüello Ron et al. “Experimental implementation of a neural network optical channel equalizer in restricted hardware using pruning and quantization”. In: *Scientific Reports* 12.1 (2022), p. 8713.

[98] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". In: *nature* 323.6088 (1986), pp. 533–536.

[99] Seb J Savory. "Digital filters for coherent optical receivers". In: *Optics express* 16.2 (2008), pp. 804–817.

[100] Seb J. Savory. "Digital Coherent Optical Receivers: Algorithms and Subsystems". In: *IEEE Journal of Selected Topics in Quantum Electronics* 16.5 (2010), pp. 1164–1179.

[101] SJ Savory et al. "Digital equalisation of 40Gbit/s per wavelength transmission over 2480km of standard fibre without optical dispersion compensation". In: *2006 European Conference on Optical Communications*. IEEE. 2009, pp. 1–2.

[102] Jürgen Schmidhuber. "Deep learning in neural networks: An overview". In: *Neural networks* 61 (Jan. 2015), pp. 85–117.

[103] M. I. Schwartz et al. "Atlanta fiber system experiment: The chicago lightwave communications project". In: *The Bell System Technical Journal* 57.6 (1978), pp. 1881–1888.

[104] Milad Sefidgaran and Mansoor Yousefi. "Lower bound on the capacity of the continuous-space SSFM model of optical fiber". In: *IEEE Transactions on Information Theory* 68.4 (2021), pp. 2460–2478.

[105] John M Senior and M Yousif Jamro. *Optical fiber communications: principles and practice*. Pearson Education, 2009.

[106] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

[107] W. Shieh, H. Bao, and Y. Tang. "Coherent optical OFDM: theory and design". In: *Opt. Express* 16.2 (Jan. 2008), pp. 841–859.

[108] Alison E. Shortt, Thomas J. Naughton, and Bahram Javidi. "A companding approach for nonuniform quantization of digital holograms of three-dimensional objects". In: *Opt. Exp.* 14.12 (June 2006), pp. 5129–5134.

[109] Oleg Sidelnikov, Alexey Redyuk, and Stylianos Sygletos. "Equalization performance and complexity analysis of dynamic deep neural networks in long haul transmission systems". In: *Opt. Express* 26.25 (Dec. 2018), pp. 32765–32776.

[110] Oleg Sidelnikov et al. “Advanced Convolutional Neural Networks for Nonlinearity Mitigation in Long-Haul WDM Transmission Systems”. In: *IEEE J. Lightw. Technol.* 39.8 (Apr. 2021), pp. 2397–2406.

[111] Han Sun, Kuang-Tsan Wu, and Kim Roberts. “Real-time measurements of a 40 Gb/s coherent system”. In: *Optics express* 16.2 (2008), pp. 873–879.

[112] B Swanson and G Gilder. “Estimating the exaflood: The impact of video and rich media on the internet?? a zetabyte? of data by 2015”. In: *Discovery Institute Report* (2008).

[113] Yoshiaki Tamura et al. “The first 0.14-dB/km loss optical fiber and its impact on submarine transmission”. In: *Journal of Lightwave Technology* 36.1 (2018), pp. 44–49.

[114] Sergey Ten. “Ultra Low-loss Optical Fiber Technology”. In: *Optical Fiber Communication Conference*. Optica Publishing Group, 2016, Th4E.5.

[115] Roy Gerardus Henricus van Uden. “MIMO digital signal processing for optical spatial division multiplexed transmission systems”. PhD thesis. Ph. D. dissertation, Dept. Elect. Eng., Eindhoven Univ. Technol., Eindhoven, 2014.

[116] Kuan Wang et al. “HAQ: Hardware-Aware Automated Quantization With Mixed Precision”. In: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* June 2019.

[117] Kuan Wang et al. “HAQ: Hardware-Aware Automated Quantization With Mixed Precision”. In: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* June 2019.

[118] S Watanabe and T Chikama. “Cancellation of four-wave mixing in multichannel fibre transmission by midway optical phase conjugation”. In: *Electronics Letters* 30.14 (1994), pp. 1156–1157.

[119] Peter J. Winzer. “From first fibers to mode-division multiplexing (Invited Paper)”. In: *Chin. Opt. Lett.* 14.12 (Dec. 2016), p. 120002.

[120] Peter J. Winzer and David T. Neilson. “From Scaling Disparities to Integrated Parallelism: A Decathlon for a Decade”. In: *Journal of Lightwave Technology* 35.5 (2017), pp. 1099–1115.

[121] Peter J. Winzer, David T. Neilson, and Andrew R. Chraplyvy. “Fiber-optic transmission and networking: the previous 20 and the next 20 years”. In: *Opt. Express* 26.18 (Sept. 2018), pp. 24190–24239.

[122] Tianhua Xu et al. “Chromatic dispersion compensation in coherent transmission system using digital filters”. In: *Optics express* 18.15 (2010), pp. 16243–16257.

[123] Kohei Yamamoto. “Learnable Companding Quantization for Accurate Low-Bit Neural Networks”. In: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* June 2021, pp. 5029–5038.

[124] Kohei Yamamoto. “Learnable companding quantization for accurate low-bit neural networks”. In: *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* 2021, pp. 5029–5038.

[125] Amnon Yariv, Dan Fekete, and David M. Pepper. “Compensation for channel dispersion by nonlinear optical phase conjugation”. In: *Opt. Lett.* 4.2 (Feb. 1979), pp. 52–54.

[126] Hao Ye, Geoffrey Ye Li, and Biing-Hwang Juang. “Deep Learning Based End-to-End Wireless Communication Systems Without Pilots”. In: *IEEE Trans. Cogn. Commun. Netw* 7.3 (2021), pp. 702–714.

[127] Penghang Yin et al. “Understanding straight-through estimator in training activation quantized neural nets”. In: *The Int. Conf. Learning Rep.* May 2019, pp. 1–30.

[128] Mansoor I Yousefi. “The asymptotic capacity of the optical fiber”. In: *arXiv preprint arXiv:1610.06458* (2016).

[129] Mansoor I Yousefi. “The Kolmogorov–Zakharov model for optical fiber communication”. In: *IEEE Transactions on Information Theory* 63.1 (2016), pp. 377–391.

[130] Yichuan Yu et al. “80 Gb/s ETDM transmitter with a traveling-wave electroabsorption modulator”. In: *OFC/NFOEC Technical Digest. Optical Fiber Communication Conference, 2005.* Vol. 3. IEEE. 2005, 3–pp.

[131] Chiyuan Zhang et al. “Understanding deep learning (still) requires rethinking generalization”. In: *Communications of the ACM* 64.3 (2021), pp. 107–115.

[132] Aojun Zhou et al. “Incremental network quantization: Towards lossless CNNs with low-precision weights”. In: *The Int. Conf. Learning Rep.* Apr. 2017, pp. 1–24.

[133] Shuchang Zhou et al. *DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients*. 2018. arXiv: 1606.06160.

[134] Xiaotian Zhu, Wengang Zhou, and Houqiang Li. "Adaptive layerwise quantization for deep neural network compression". In: *2018 IEEE Int. Conf. Multimedia Expo (ICME)*. IEEE. 2018, pp. 1–6.

ECOLE
DOCTORALE

Titre : Quantification des réseaux de neurones pour l'égalisation dans les communications par fibre optique

Mots clés : Quantification,réseaux de neurones, fibre optique,traitement numérique du signal

Résumé : L'avènement de la détection cohérente a ouvert la voie à la compensation des effets liés à la propagation dans les fibres optiques en utilisant le traitement numérique du signal ("DSP"). Alors que les effets linéaires, tels que la dispersion chromatique et la dispersion modale de polarisation, peuvent être compensées efficacement, la compensation des distorsions non linéaires reste aujourd'hui un défi compte-tenu des complexités d'implémentation.
Dans ce travail, nous considérons les réseaux de neurones ("NN") pour l'égalisation dans la transmission par fibre optique à double polarisation. Par rapport aux égaliseurs conventionnels tels que la rétropropagation numérique ("DBP"), les NN ne nécessitent pas d'informations sur l'état du canal, et peuvent atténuer les dégradations du signal avec une moindre complexité. Nous proposons un certain nombre d'algorithmes de quantification "post-training" et "training-aware" pour représenter les poids et les activations du NN en quelques bits, ceci afin de réduire la complexité de calcul, l'espace mémoire et la consommation d'énergie du DSP. Une analyse de performance et de complexité montrent que les algorithmes proposés surpassent les algorithmes d'égalisation linéaire et DBP dans plusieurs expériences de transmission.
Cette thèse est réalisée dans le cadre du projet H2020 MSCA-ITN-EID REAL-NET, financée par la Commission Européenne (en collaboration avec le partenaire industriel, Infinera Corporation, en Allemagne et au Portugal).

Title : Quantization of Neural Network Equalizers in Optical Fiber Transmission Experiments

Keywords : Quantization,neural networks,optical fiber,digital signal processing

Abstract : The advent of the coherent detection paved the away for the compensation of the transmission effects in optical fiber using the digital signal processing (DSP). While the linear effects, such as the chromatic dispersion and polarization-induced impairments, can be efficiently compensated with DSP, the compensation of the nonlinear distortions remains challenging.
In this work, we consider neural networks (NNs) for equalization in dual-polarization optical fiber transmission. Compared to the conventional equalizers such as the digital back-propagation (DBP), NNs do not require the channel state information, and may mitigate the impairments with lower complexity. We propose a number of post-training and training-aware quantization algorithms for representing the weights and activations of the NN in few bits, in order to reduce the computational complexity, memory requirement and energy consumption of the DSP. A performance and complexity analysis shows that the proposed algorithms outperform the linear equalization and DBP in several transmission experiments.
This thesis is carried out in the framework of the H2020 MSCA-ITN-EID REAL-NET project, funded by the European Commission (in collaboration with the industry partner, Infinera Corporation, in Germany and Portugal).

Institut Polytechnique de Paris
91120 Palaiseau, France