Non-Linearity in Wireless Communications and Deep Learning

by

Bishwash Khanal

Bachelor Thesis in Electrical and Computer Engineering

Submission: June 27, 2020
Supervisor: Prof. Dr. Giuseppe T. F. de Abreu

Jacobs University Bremen | Department of Computer Science and Electrical Engineering

Statutory Declaration

Family Name, Given/First Name	Khanal, Bishwash
Matriculation number	30001397
What kind of thesis are you submitting: Bachelor-, Master- or PhD-Thesis	Bachelor-Thesis

English: Declaration of Authorship

I hereby declare that the thesis submitted was created and written solely by myself without any external support. Any sources, direct or indirect, are marked as such. I am aware of the fact that the contents of the thesis in digital form may be revised with regard to usage of unauthorized aid as well as whether the whole or parts of it may be identified as plagiarism. I do agree my work to be entered into a database for it to be compared with existing sources, where it will remain in order to enable further comparisons with future theses. This does not grant any rights of reproduction and usage, however.

This document was neither presented to any other examination board nor has it been published.

Deutsch: Erklärung der Autorenschaft (Urheberschaft)

Ich erkläre hiermit, dass die vorliegende Arbeit ohne fremde Hilfe ausschließlich von mir erstellt und geschrieben worden ist. Jedwede verwendeten Quellen, direkter oder indirekter Art, sind als solche kenntlich gemacht worden. Mir ist die Tatsache bewusst, dass der Inhalt der Thesis in digitaler Form geprüft werden kann im Hinblick darauf, ob es sich ganz oder in Teilen um ein Plagiat handelt. Ich bin damit einverstanden, dass meine Arbeit in einer Datenbank eingegeben werden kann, um mit bereits bestehenden Quellen verglichen zu werden und dort auch verbleibt, um mit zukünftigen Arbeiten verglichen werden zu können. Dies berechtigt jedoch nicht zur Verwendung oder Vervielfältigung.

Diese Arbeit wurde noch keiner anderen Prüfungsbehörde vorgelegt noch wurde sie bisher veröffentlicht.

Date, Signature

Acknowledgement

I am immensely thankful to my supervisor Prof. Dr. Giuseppe T. F. de Abreu, who guided me throughout this thesis. He taught me the scientific methods and got me familiar with the project plan. His constant feedback was of utmost importance.

I would like to dedicate my gratitude to my family who always motivated and encouraged me in my career and provided love and support.

Abstract

Deep Learning has been known to optimize real world problems with robust mathematical models. One of such applications could be modeling a physical layer for Wireless Communcations using Neural Networks (NN). This thesis proposes the use of Autoencoder to represent a simple communications system. During transmission of a signal, non-linearity is introduced due to hardware impairments and amplification of signal. Also, self-interference caused from full-duplex communication system adds non-linearity. In these cases, the constellation from conventional modulation schemes are not optimal.
This thesis will focus on modeling such distortions as non-linear Additive-White Gaussian Noise (AWGN) channel and finding the optimal constellation to overcome this non-linearity using deep learning methods. The quality of the NN is assessed with the help of Bit Error Rate (BER) plots. We see that the Autoencoder is able to match the performance of standard constellation in linear noise while it outperforms those constellation in non-linear noise. The results are then verified using analysis on constellation of simple Pulse-amplitude Modulation (PAM).

Keywords: Wireless Communications, Machine Learning, Deep Learning, Digital Modulation, Autoencoders, Physical Layer.

Contents

List of Abbreviations

AWGN	Additive-White Gaussian Noise
ANN	Artificial Neural Network
AEC	Autoencoder
BER	Bit Error Rate
CDMA	Code-Division Multiple Access
GCS	Geometric Constellation Shaping
MLP	Multilayer Perceptron
MIMO	Multiple Input Multiple Output
NN	Neural Network
PSK	Phase Shift Key
PCS	Probabilistic Constellation Shaping
PAM	Pulse-Amplitude Modulation
QAM	Quadrature Amplitude Modulation
RF	Radio-Frequency
SNR	Signal-to-Noise Ratio
SGD	Stochastic Gradient Method

1 Introduction

The evolution of technology has introduced many obstacles and consequently many methods to tackle them. Deep learning is one of those tools which optimizes a problem by learning on existing data to predict the outcome for a previously unsolved problem. Pattern recognition, classification problem, function approximation are few sectors where Deep Learning has had a significant impact. They are used to analyse a huge cluster of data that has helped in the rapid development in the fields like self-driving, automation, computer vision, virtual assistant and many more. Communications is one of those sectors where deep learning has been used to optimize network parameters. From multi-user detection in code-division multiple access (CDMA) [1] to optimization of traffic routing in communication network [2], the neural network has been used in communications from early days of its introduction.

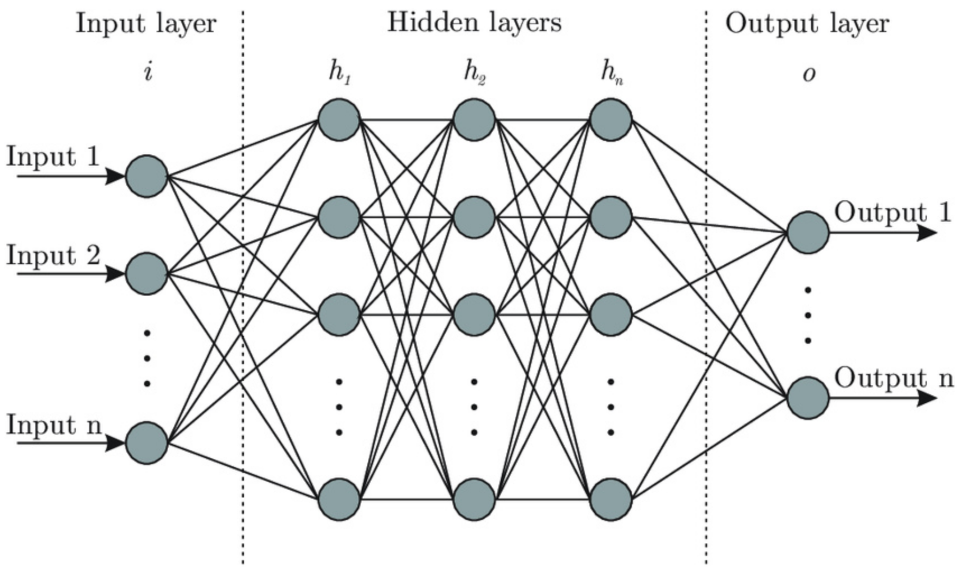


Figure 1: General Structure of a neural network [3].

Lately, due to the development of complex models for communications, conventional digital modulation schemes are losing performance. For instance, for long range transmission of signal, fiber-optic communication is used. Optical fiber are considered to be a non-linear medium [4]. And over a large distance, it introduces non-linear distortion in the transmitted signal. So, to overcome this distortion, the concept of constellation shaping has been introduced [5].

The basic idea for constellation shaping is to modify the transmitted symbols for the existing modulation schemes such that the non-linear effects are mitigated. Based on the constellation obtained, there are two types of constellation shaping i.e. Probabilistic Constellation Shaping (PCS) and Geometric Constellation Shaping (GCS). The probabilities of transmission of symbols are optimized in PCS while the whole geometry of constellation are changed in the GCS [6].

The problem of non-linearity exists in Wireless Communications as well. Due to power amplifiers and non-linear property of RF antenna, symbols are distorted in the transmitter itself [7]. And lately, the concept of full-duplex has been introduced in Wireless Communications. In full-duplex system, a communicator transmits and receives signal simultaneously through the same channel. This introduces additional non-linearity in the received signal as its own transmission is also received as interference and sometime interference is more powerful than the actual signal. To mitigate this problem, different methods of constellation shaping have been applied to obtain gain over the standard modulation. Limori et al. [8] used rate maximization in AWGN channel to obtain PS constellation while bit-interleaved polar coded modulation (BIPCM) was introduced in Zhou et al. [9]. In this thesis, deep learning is proposed to overcome this issue.

As Neural Network are capable of learning features from existing data, Deep Learning can be used to optimize the design of constellation. O'Shea et al. [10] has applied deep learning to obtain GCS for a normal AWGN channel. Feedforward NN like Autoencoder can be used as it shares many similarities with communications system. The communications channel can be modeled as a Gaussian layer. Since NN tries to learn unique representation of its input, geometrically shaped constellation are obtained from its encoder. This thesis aims to model the channel layer as non-linear AWGN channel to learn constellation that adapts to the non-linearity over such channel to gain performance over conventional modulation schemes.

2 Motivation of the Research

2.1 Analysis on 4PAM

An analysis was done with 4PAM constellation to compare performance of optimized constellation against standard constellation under non-linear Gaussian noise. Since 4PAM is one-dimensional constellation scheme, the Gaussian (AWGN) applied to it have has only one variable and the complexity is low for analysis. Figure 2 shows how non-linear Gaussian is applied on 4PAM constellation. Here the mean of the Gaussian are $\mu = a, b$ and variance (σ^2) is $\sigma_0^2 + \beta x^2$ where σ_0^2 is variance of standard Gaussian, β is intrinsic noise level of the front-end transmit chain and is usually small ($0 < \beta \ll 1$) [8] and $x = a, b$ are parameters which introduces non-linearity.

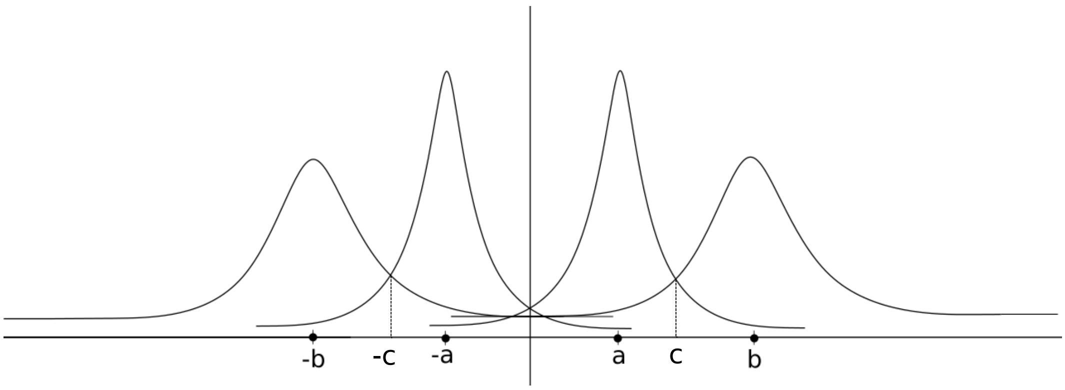


Figure 2: 4PAM under non-linear Gaussian Noise.

From the Figure 2 , we can see that the following areas of the Gaussian are equal which are also error probabilities for detection of a and b. $\pm c$ is the point of intersection of two Gaussian.

$$\frac{1}{\sqrt{2\pi(\sigma_0^2 + \beta a^2)}} \int_{-\infty}^{-c} e^{\frac{-(x+a)^2}{2(\sigma_0^2+\beta a^2)}} dx \approx \frac{1}{\sqrt{2\pi(\sigma_0^2 + \beta b^2)}} \int_{-c}^{0} e^{\frac{-(x+b)^2}{2(\sigma_0^2+\beta b^2)}} dx$$
$$\approx \frac{1}{\sqrt{2\pi(\sigma_0^2 + \beta a^2)}} \int_{0}^{c} e^{\frac{-(x+a)^2}{2(\sigma_0^2+\beta a^2)}} dx \quad (1)$$

Converting these Integrals into the standard Gaussian Integrals with zero mean and unit variance through substitution, we get,

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{-c+a}{\sqrt{\sigma_0^2+\beta a^2}}} e^{\frac{-y^2}{2}} dy \approx \frac{1}{\sqrt{2\pi}} \int_{\frac{-c+b}{\sqrt{\sigma_0^2+\beta b^2}}}^{\frac{b}{\sqrt{\sigma_0^2+\beta b^2}}} e^{\frac{-y^2}{2}} dy$$
$$\approx \frac{1}{\sqrt{2\pi}} \int_{\frac{a}{\sqrt{\sigma_0^2+\beta a^2}}}^{\frac{c+a}{\sqrt{\sigma_0^2+\beta a^2}}} e^{\frac{-y^2}{2}} dy \quad (2)$$

Again, converting them into standard Q-function notation, we get a relation,

$$Q\left(\frac{c-a}{\sqrt{\sigma_0^2+\beta a^2}}\right) = Q\left(\frac{b-c}{\sqrt{\sigma_0^2+\beta b^2}}\right) - Q\left(\frac{b}{\sqrt{\sigma_0^2+\beta b^2}}\right)$$
$$= Q\left(\frac{a}{\sqrt{\sigma_0^2+\beta a^2}}\right) - Q\left(\frac{c+a}{\sqrt{\sigma_0^2+\beta a^2}}\right) \quad (3)$$

From this relation, the following equalities were observed.

$$Q\left(\frac{c-a}{\sqrt{\sigma_0^2+\beta a^2}}\right) = Q\left(\frac{b-c}{\sqrt{\sigma_0^2+\beta b^2}}\right) - Q\left(\frac{b}{\sqrt{\sigma_0^2+\beta b^2}}\right) \quad (4)$$

$$Q\left(\frac{c-a}{\sqrt{\sigma_0^2+\beta a^2}}\right) = Q\left(\frac{a}{\sqrt{\sigma_0^2+\beta a^2}}\right) - Q\left(\frac{c+a}{\sqrt{\sigma_0^2+\beta a^2}}\right) \quad (5)$$

$$Q\left(\frac{b-c}{\sqrt{\sigma_0^2+\beta b^2}}\right) - Q\left(\frac{b}{\sqrt{\sigma_0^2+\beta b^2}}\right)$$
$$= Q\left(\frac{a}{\sqrt{\sigma_0^2+\beta a^2}}\right) - Q\left(\frac{c+a}{\sqrt{\sigma_0^2+\beta a^2}}\right) \quad (6)$$

On further simplifying these relations and using Supertight bound on Q-functions from [11], following equation was obtained for the first relation.

$$\frac{1}{50}e^{\frac{-(c-a)^2}{\sigma_0^2+\beta a^2}} + \frac{1}{2(\frac{c-a}{\sqrt{\sigma_0^2+\beta a^2}}+1)}e^{\frac{-(c-a)^2}{2(\sigma_0^2+\beta a^2)}} = \frac{1}{50}e^{\frac{-(b-c)^2}{\sigma_0^2+\beta b^2}}$$
$$+\frac{1}{2(\frac{b-c}{\sqrt{\sigma_0^2+\beta b^2}}+1)}e^{\frac{-(b-c)^2}{2(\sigma_0^2+\beta b^2)}} - \frac{1}{50}e^{\frac{-b^2}{\sigma_0^2+\beta b^2}} - \frac{1}{2(\frac{b}{\sqrt{\sigma_0^2+\beta b^2}}+1)}e^{\frac{-b^2}{2(\sigma_0^2+\beta b^2)}} \quad (7)$$

For the second equality, we got,

$$\frac{1}{50}e^{\frac{-(c-a)^2}{\sigma_0^2+\beta a^2}} + \frac{1}{2(\frac{c-a}{\sqrt{\sigma_0^2+\beta a^2}}+1)}e^{\frac{-(c-a)^2}{2(\sigma_0^2+\beta a^2)}} = \frac{1}{50}e^{\frac{-a^2}{\sigma_0^2+\beta a^2}}$$
$$+\frac{1}{2(\frac{a}{\sqrt{\sigma_0^2+\beta a^2}}+1)}e^{\frac{-a^2}{2(\sigma_0^2+\beta a^2)}} - \frac{1}{50}e^{\frac{-(c+a)^2}{\sigma_0^2+\beta a^2}} - \frac{1}{2(\frac{c+a}{\sqrt{\sigma_0^2+\beta a^2}}+1)}e^{\frac{-(c+a)^2}{2(\sigma_0^2+\beta a^2)}} \quad (8)$$

And finally, following equation was obtained for third relation.

$$\frac{1}{50}e^{\frac{-(b-c)^2}{\sigma_0^2+\beta b^2}} + \frac{1}{2(\frac{b-c}{\sqrt{\sigma_0^2+\beta b^2}}+1)}e^{\frac{-(b-c)^2}{2(\sigma_0^2+\beta b^2)}} - \frac{1}{50}e^{\frac{-b^2}{\sigma_0^2+\beta b^2}} - \frac{1}{2(\frac{b}{\sqrt{\sigma_0^2+\beta b^2}}+1)}e^{\frac{-b^2}{2(\sigma_0^2+\beta b^2)}}$$

$$= \frac{1}{50}e^{\frac{-a^2}{\sigma_0^2+\beta a^2}} + \frac{1}{2(\frac{a}{\sqrt{\sigma_0^2+\beta a^2}}+1)}e^{\frac{-a^2}{2(\sigma_0^2+\beta a^2)}} - \frac{1}{50}e^{\frac{-(c+a)^2}{\sigma_0^2+\beta a^2}} - \frac{1}{2(\frac{c+a}{\sqrt{\sigma_0^2+\beta a^2}}+1)}e^{\frac{-(c+a)^2}{2(\sigma_0^2+\beta a^2)}}$$
$$(9)$$

Now, we know that the average energy of the constellation should be unity i.e.

$$\frac{2a^2}{4} + \frac{2b^2}{4} = 1 \tag{10}$$

$$b = \sqrt{2 - a^2} \tag{11}$$

Substituting $\sigma_0^2 + \beta a^2 = \sigma_a^2$, $\sigma_0^2 + \beta b^2 = \sigma_b^2$, and replacing the value of b, the equations can be rearranged as follow.

$$\frac{1}{50}e^{\frac{-(c-a)^2}{\sigma_a^2}} - \frac{1}{50}e^{\frac{-(\sqrt{2-a^2}-c)^2}{\sigma_b^2}} + \frac{1}{50}e^{\frac{-(2-a^2)}{\sigma_b^2}} + \frac{1}{2(\frac{c-a}{\sigma_a}+1)}e^{\frac{-(c-a)^2}{2\sigma_a^2}} - \frac{1}{2(\frac{\sqrt{2-a^2}-c}{\sigma_b}+1)}e^{\frac{-(\sqrt{2-a^2}-c)^2}{2\sigma_b^2}} + \frac{1}{2(\frac{\sqrt{2-a^2}}{\sigma_b}+1)}e^{\frac{-(2-a^2)}{2\sigma_b^2}} = 0 \tag{12}$$

$$\frac{1}{50}e^{\frac{-(c-a)^2}{\sigma_a^2}} - \frac{1}{50}e^{\frac{-a^2}{\sigma_a^2}} + \frac{1}{50}e^{\frac{-(c+a)^2}{\sigma_a^2}} + \frac{1}{2(\frac{c-a}{\sigma_a}+1)}e^{\frac{-(c-a)^2}{2\sigma_a^2}} - \frac{1}{2(\frac{a}{\sigma_a}+1)}e^{\frac{-a^2}{2\sigma_a^2}} + \frac{1}{2(\frac{c+a}{\sigma_a}+1)}e^{\frac{-(c+a)^2}{2\sigma_a^2}} = 0 \tag{13}$$

$$\frac{1}{50}e^{\frac{-(\sqrt{2-a^2}-c)^2}{\sigma_b^2}} - \frac{1}{50}e^{\frac{-(2-a^2)}{\sigma_b^2}} + \frac{1}{50}e^{\frac{-(c+a)^2}{\sigma_a^2}} - \frac{1}{50}e^{\frac{-a^2}{\sigma_a^2}} + \frac{1}{2(\frac{\sqrt{2-a^2}-c}{\sigma_b}+1)}e^{\frac{-(\sqrt{2-a^2}-c)^2}{2\sigma_b^2}} - \frac{1}{2(\frac{\sqrt{2-a^2}}{\sigma_b}+1)}e^{\frac{-(2-a^2)}{2\sigma_b^2}} - \frac{1}{2(\frac{a}{\sigma_a}+1)}e^{\frac{-a^2}{2\sigma_a^2}} + \frac{1}{2(\frac{c+a}{\sigma_a}+1)}e^{\frac{-(c+a)^2}{2\sigma_a^2}} = 0 \tag{14}$$

Then, Equations 12, 13 and 14 were plotted as a function of a and c for a fixed β and σ_0. Since Equation 13 is independent of b, it was used as a objective function after taking square to omit negative values. To constellation to have unit energy in standard 4PAM, a must be equal to $\frac{1}{\sqrt{5}}$ while c should be $\frac{2}{\sqrt{5}}$. Considering this fact, the range of a was chosen to be $0 < a < \frac{1}{\sqrt{5}}$ and c as $a < c < \frac{2}{\sqrt{5}}$ and Equation 15 was plotted as shown in Figure 3.

$$\left(\frac{1}{50}e^{\frac{-(c-a)^2}{\sigma_a^2}} - \frac{1}{50}e^{\frac{-a^2}{\sigma_a^2}} + \frac{1}{50}e^{\frac{-(c+a)^2}{\sigma_a^2}} + \frac{1}{2(\frac{c-a}{\sigma_a}+1)}e^{\frac{-(c-a)^2}{2\sigma_a^2}} - \frac{1}{2(\frac{a}{\sigma_a}+1)}e^{\frac{-a^2}{2\sigma_a^2}} + \frac{1}{2(\frac{c+a}{\sigma_a}+1)}e^{\frac{-(c+a)^2}{2\sigma_a^2}}\right)^2 = 0 \tag{15}$$

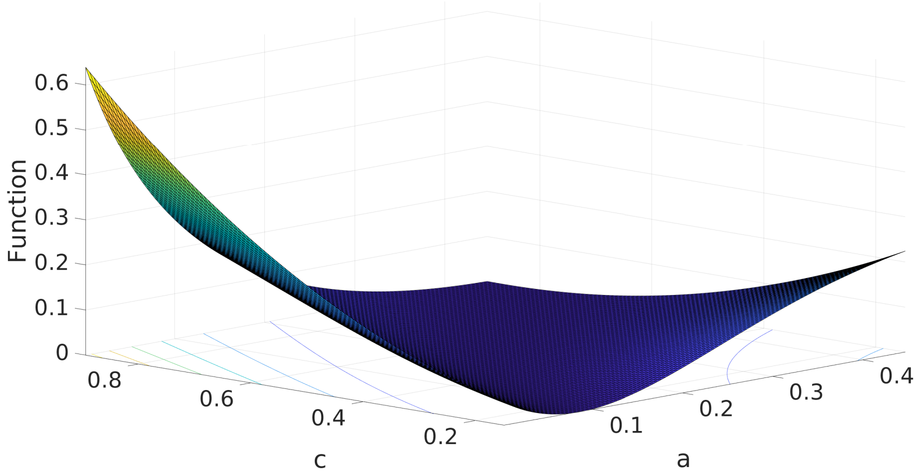


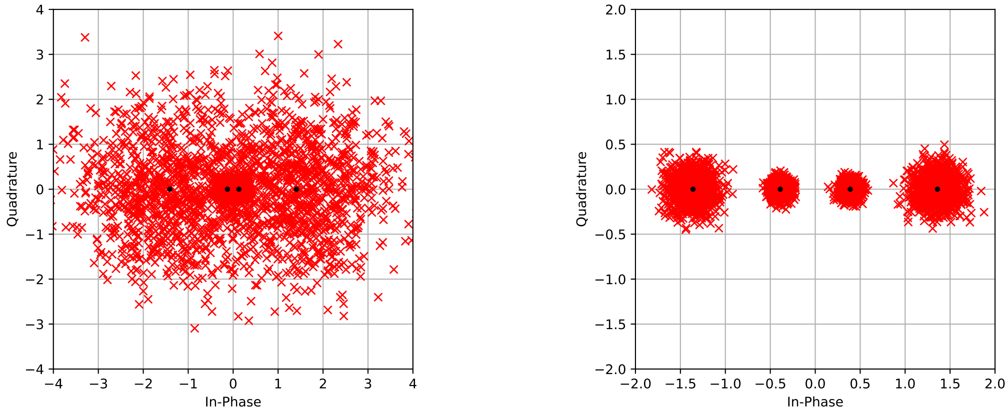
Figure 3: Surface structure of Equation 15 ($\beta = 1$, SNR = 5dB).

As seen in Figure 3, the problem was found to be convex. Hence, Steepest Decent Method Algorithm was used to find the optimal a and c for given non-linearity (β). For initial guess of a and c, brute-force optimization method was used on the curve above. The following table shows the calculated a and c from algorithm for given β.

β	a	c
1	0.0227	0.2747
0.5	0.1278	0.3761
0.1	0.3072	0.6297
0.01	0.3891	0.7810
0	0.4389	0.8941

2.2 Optimized Constellation

Figure 4 shows the constellation plots of analytically optimized 4PAM with the respective non-linear noise.



(a) $\beta = 0.5$, SNR = 20dB

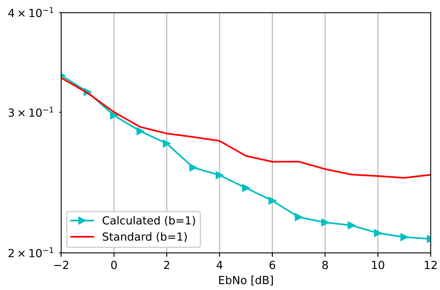
(b) $\beta = 0.01$, SNR = 20dB

Figure 4: Analysed Optimized constellation for 4PAM in non-linear noises.

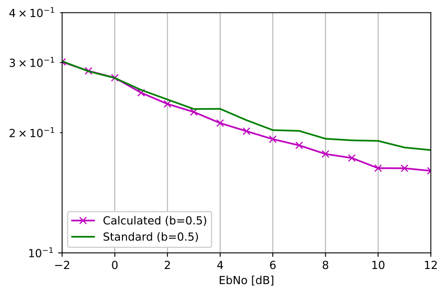
The *red cloud* around the constellation (*black points*) represents the noise. SNR (Signal-to-Noise Ratio) of 20dB was chosen in order to give a clear visualization that the noise was in fact non-linear. As non-linearity (β) increases, a was found to move closer to origin while b moves away which is an expected observation. The cloud of non-linear noise is strong in b, hence moving b away from a increases the possibility of detection of both constellation. To verify this, BER of the optimized constellation was compared against that of standard constellation for 4PAM.

2.3 Comparison of Bit Error Rate

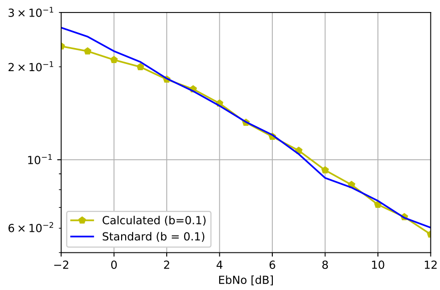
Figure 5 gives the comparison of performance of optimized constellation against standard constellation.



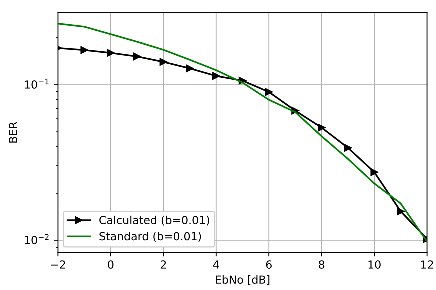
(a) $\beta = 1$



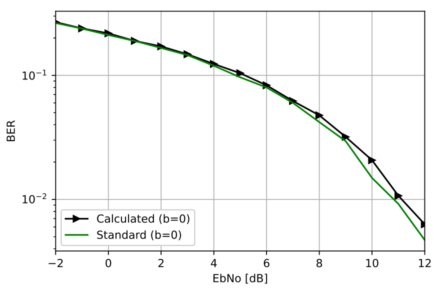
(b) $\beta = 0.5$



(c) $\beta = 0.1$



(d) $\beta = 0.01$



(e) $\beta = 0$

Figure 5: BER comparison of optimized and standard 4PAM under non-linear noise.

From the plots in Figure 5, we can observe that at high non-linearity (i.e. $\beta = 1$ and 0.5), the BER of standard and calculated constellation is similar but as SNR increases, calculated constellation outperforms the standard one. As, non-linearity decreases, at low SNR, calculated constellation outperforms the standard one while performing similar in high SNR as expected. For no non-linearity ($\beta = 0$), they perform exactly the same as expected which can be observed in Figure 5e.

2.4 Analysis on 16QAM

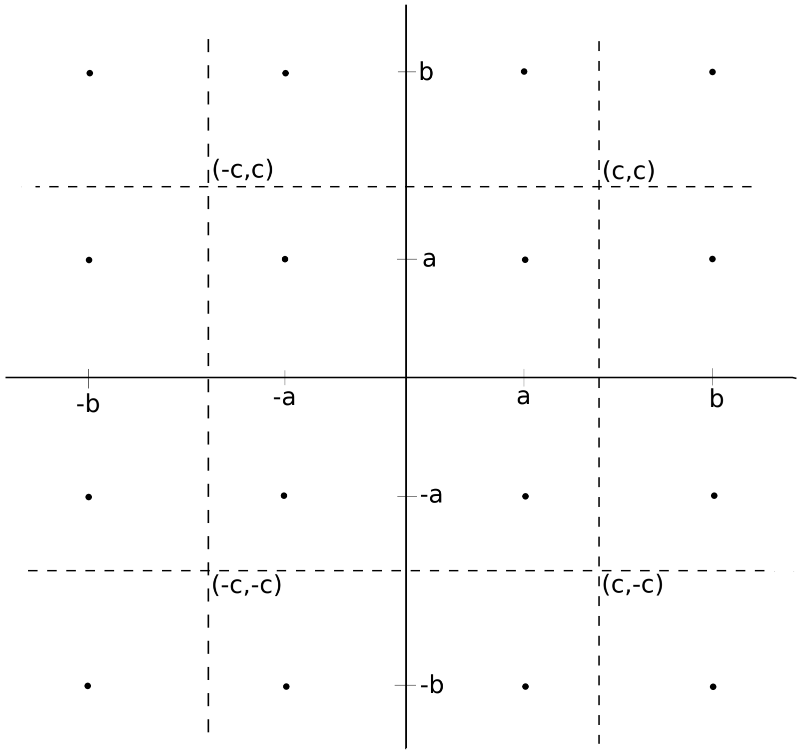


Figure 6: Constellation Distribution for 16QAM under non-linear noise.

In the figure, we can see the constellation distribution for 16QAM (Quadrature Amplitude Modulation) under non-linear noise. Here lines $x, y = \pm c$ denotes the intersection of different 2D Gaussian. Intuitively, we can solve the equalities obtained from the following integrals as we did in 4PAM to obtain optimal a and b.

$$\frac{1}{2\pi\sigma_a\sigma_b\sqrt{(1-\rho^2)}}\int_c^\infty\int_{-\infty}^{-c} e^{-\frac{1}{2(1-\rho^2)}\left[\frac{(x+a)^2}{\sigma_a^2}+\frac{(y-b)^2}{\sigma_b{}^2}-2\rho\frac{(x+a)(y-b)}{\sigma_a\sigma_b}\right]}dxdy$$

$$\approx \frac{1}{2\pi\sigma_b\sigma_b\sqrt{(1-\rho^2)}}\int_c^\infty\int_{-c}^{0} e^{-\frac{1}{2(1-\rho^2)}\left[\frac{(x+b)^2}{\sigma_b^2}+\frac{(y-b)^2}{\sigma_b{}^2}-2\rho\frac{(x+b)(y-b)}{\sigma_b\sigma_b}\right]}dxdy \approx \cdots \quad (16)$$

where $\sigma_a = \sqrt{\sigma_0^2 + \beta a^2}$, $\sigma_b = \sqrt{\sigma_0^2 + \beta b^2}$ and ρ is correlation coefficient between x and y. The problem above can be solved but is highly complex as compared to 4PAM. It has additional variables and two integrals. Hence, we can solve the problem of non-linearity in 16QAM with the help of neural networks as analysis is complex. The following sections contain a detailed description on how constellation learned for 16QAM from NN can outperform standard constellation under non-linear noise.

3 Description of the Research

This section provides a general structure of Autoencoder and communications system.

3.1 Deep Learning

Deep Learning is one of the methods of machine learning based on Artificial Neural Network (ANN). ANN has multiple hidden layers and learns higher dimensional features from input [12].

A Multilayer Perceptron (MLP) i.e. feed-forward neural network $\mathcal{N}$ with K hidden layers gives a mapping $\mathcal{N} : \mathbb{R}^{L_0} \to \mathbb{R}^{L_K}$ for input vector $a^0 \in \mathbb{R}^{L_0}$ to output vector $a^K \in \mathbb{R}^{L_K}$ by iterative process

$$a^m = \mathcal{N}(a^{m-1}, \theta^m), \qquad m = 1, ..., K \tag{17}$$

where $\theta = \{\theta^1, ..., \theta^K\}$ are parameters of hidden layers. For $\theta^m = \{W^m, b^m\}$, W^m is synaptic connection weight matrix for links between layers $m-1$ and m and b^m is a vector of bias weights. A given mth layer is called Dense layer and is given by form

$$\mathcal{N}(a^{m-1}, \theta^m) = \sigma(W^m a^{m-1} + b^m) \tag{18}$$

where $\sigma(\cdot)$ is an activation function. The activation functions are generally non-linear which improve expressive power of NN. It gives advantage of having multiple layers together [13]. Some of the activation function that are used in our NN are mentioned in Table 1.

Name	$\sigma(u_i)$	Range
ReLU (rectifier)	$max\{0, u_i\}$	$[0, \infty)$
Linear	u_i	$(-\infty, \infty)$
Softmax	$\frac{e^{u_i}}{\sum_{j=1,...,k} e^{u_j}}$	$(0, 1)$
Sigmoid	$\frac{1}{1+e^{-u_i}}$	$(0, 1)$
Hyperbolic tan	$\tanh(u_i)$	$(-1, 1)$

Table 1: List of Activation functions.

The whole process of training a NN is an optimization task, where the goal is to find function $\mathcal{N}$ which minimizes the loss

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} l(\hat{a}_i, \mathcal{N}(a_i)) \tag{19}$$

based on the parameters in θ with the help of loss function $l(u, v)$. Some of the loss functions are given in Table 2. Here $\hat{a}_i$ is the desired output while $\mathcal{N}(a_i)$ is the output of NN. We apply different type of optimization algorithms to find good estimate of θ. One of those algorithms is Stochastic Gradient Descent (SGD) which updates θ iteratively as shown in Equation 20.

$$\theta^n = \theta^{n-1} - \alpha \nabla L(\theta^{n-1}) \tag{20}$$

The gradient ($\nabla L(\theta^{n-1})$) in Equation 20 can be efficiently calculated through back-propagation algorithm. Here, α is the step-size of the function or otherwise known as learning rate of NN.

Name	$l(u, v)$
Categorical cross-entropy	$-\sum_{j=1,..,k} u_j log(v_j)$
MSE	$\|u - v\|_2^2$
Poisson	$v - ulog(v)$

Table 2: List of Loss functions.

3.2 Autoencoders

Autoencoder is an application of a neural network which tries to replicate its input at the output node. It is an unsupervised learning process in a sense that we do not provide any additional output data during the training process [14]. It comprises of three layers unlike other neural networks i.e. input layer, hidden layer and output layer.

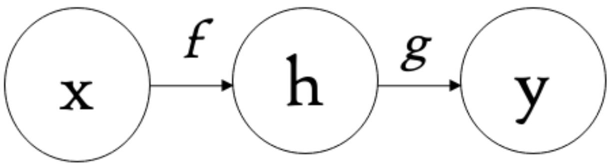


Figure 7: A schematic of basic Autoencoder.

Among the three layers, the input layer and output layer have the same dimension and the hidden layer has one or more layers. In the hidden layer, all dimensional reduction and expansion occurs. In the first phase, the dimension of each layers keeps on decreasing starting from the input layer. The layer with the least dimension is called the bottleneck or data layer because if we have to transfer or store data, we use output of this layer. Bottleneck layer also acts as a bridge between encoding and decoding layers of the neural network. After this layer, the second phase starts. During the learning process in the hidden layer h, it learns the representation of its input which may describe its properties[15]. In the second phase, the dimension is expanded to give output which is more or less same as input. Autoencoder is used in various applications like signal reconstruction, noise reduction, data compression and so on[16].

3.3 Communications system

A simplest form of communications system has a transmitter, a channel and a receiver as shown in Figure 8. From transmitter, one of the M messages $s \in \mathbb{M} = \{1, ..., M\}$ is sent to receiver through n channels. At the transmitter, transformation $f : \mathbb{M} \to \mathbb{R}^n$ is applied to s which generates transmitted signal $\mathbf{x} = f(s) \in \mathbb{R}^n$. Here, $\mathbf{x}$ represents a complex symbol of length 2 (real and imaginary). It gives the constellation mapping from the transmitter (encoder). It represents s and is optimized to be robust to the channel perturbation [17]. Optimization of this constellation is one of the motivations of this research. There are hardware imperfections which impose constraints on transmitted signal like an energy constraint $\|\mathbf{x}\|_2^2 \leq n$, a power constraint $\mathbb{E}[|x_i|^2] \leq 1 \; \forall i$ or an amplitude constraint $|x_i| \leq 1 \; \forall i$. The constellation produced should be optimized for these imperfections and channel impairments. The conditional probability $p(\mathbf{y}|\mathbf{x})$ describes the channel where $\mathbf{y} \in \mathbb{R}^n$ is the received signal. Then transformation $g : \mathbb{R}^n \to \mathbb{M}$ is applied to $\mathbf{y}$ to find estimate of s as $\tilde{s}$. $R = \frac{k}{n} \frac{bits}{channel}$ is the communication rate of the system where $k = log_2(M)$.

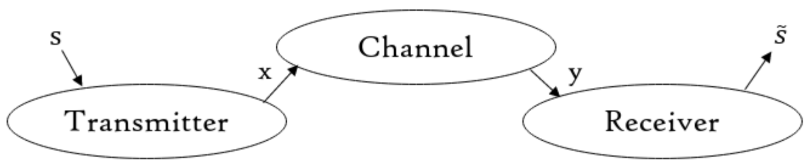


Figure 8: A simple communication system.

4 Autoencoder in Non-Linear Noise

4.1 Autoencoder Setup

The description of the Autoencoder which acts as a communications system is given in Figure 9. The Autoencoder was divided into three sections namely Transmitter which acted as a encoder, Channel with disturbances (noise), and Receiver which is the decoder part. The transmitter was set up with multiple dense layers with different activation functions. Then the layers were followed by a normalization layer (energy or power normalization). Depending on the type of normalization chosen, different shape of constellation was obtained. The input **s** was encoded as one-hot vector $\mathbf{1}_s \in \mathbb{R}^M$ (M dimensional vector with 1 in sth position and 0 elsewhere) and was fed into the transmitter.

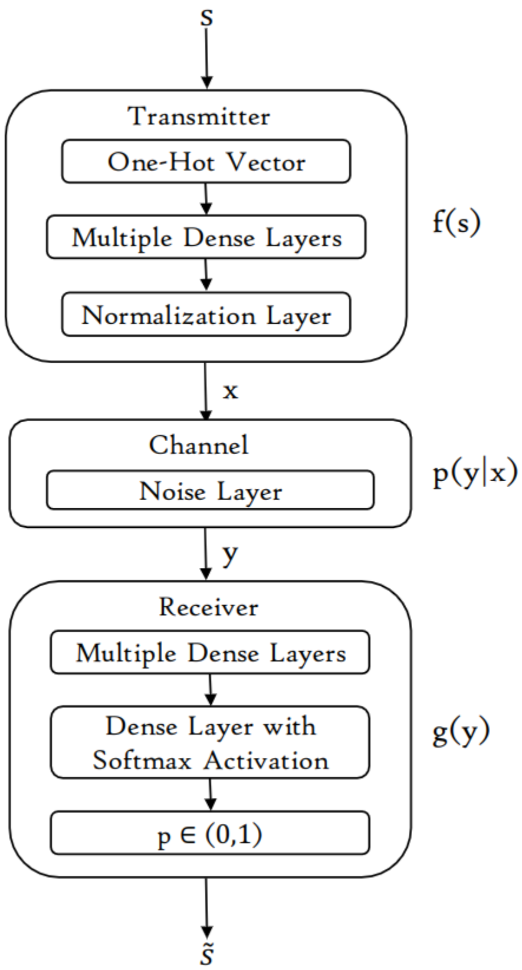


Figure 9: Autoencoder as a communications system.

The transmitter was followed by a Channel layer which includes a noise. AWGN was taken to be the noise with two different variances. First one was $\mathcal{CN}(0,\, \sigma^2)$, where $\sigma^2 = \frac{1}{2E_b/N_0}$ (E_b/N_0 is energy per bit to noise ratio), which is a linear representation of Gaussian while second one was $\mathcal{CN}(0,\, \beta|x_i|^2 + \sigma^2)$ where $|x_i|^2$ is the instantaneous power of the symbols. The second variance gave us a non-linear noise as it depends on the power of the individual symbols in a constellation. The β was varied accordingly to observe the effect of different non-linearity on the learned constellation.

In receiver, a dense layer with rectified linear as activation was followed by a softmax activated dense layer which gave us output as $\mathbf{p} \in (0,1)^M$. Then element of $\mathbf{p}$ with highest probability was considered to be the decoded message $\tilde{s}$. After training the system as Autoencoder using SGD, categorical cross-entropy loss function between 1_s and $\mathbf{p}$ was used [10] to find the optimum parameters of the neural network. A detailed layout of the Autoencoder is provided in the Table 3. Given the dimension of outputs, $(2M+1)(M+n)+2M$ parameters were trained.

Layer	**Output Dimensions**
Input	M
Dense (ReLU)	M
Dense (Linear)	n
Normalization	n
Noise	n
Dense (ReLU)	M
Dense (Softmax)	M

Table 3: Layout of layers in Autoencoder.

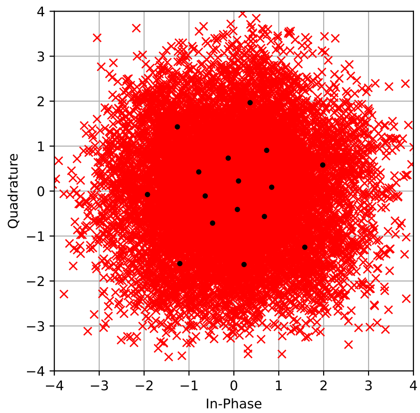
Data Generation

Data for training, validating and testing the neural network were generated as follows:

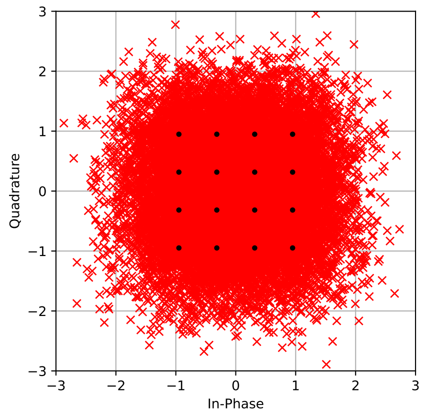
- For training, 48,000 random labels **s** were produced with the value ranging from 1 to M. Then, 32,000 row vectors with 1 in the given labels and 0 elsewhere (one-hot vectors) were created to generate 32000 arrays of length M.
- For validation, 16,000 random labels were produced to generate 15,000 arrays of one-hot vectors.
- Similarly, 16,000 labels were produced to generate 50,000 one-hot vectors of length M for testing the neural network.

4.2 Learned Constellation

The Neural Network was trained such that for the given amount of non-linearity (β), it would produce constellation for every SNR. Few plots of learned and standard constellation can be seen below as well as in Appendix II for different level of noises.



(a) 16AEC ($\beta = 0.1$, SNR = 5dB)



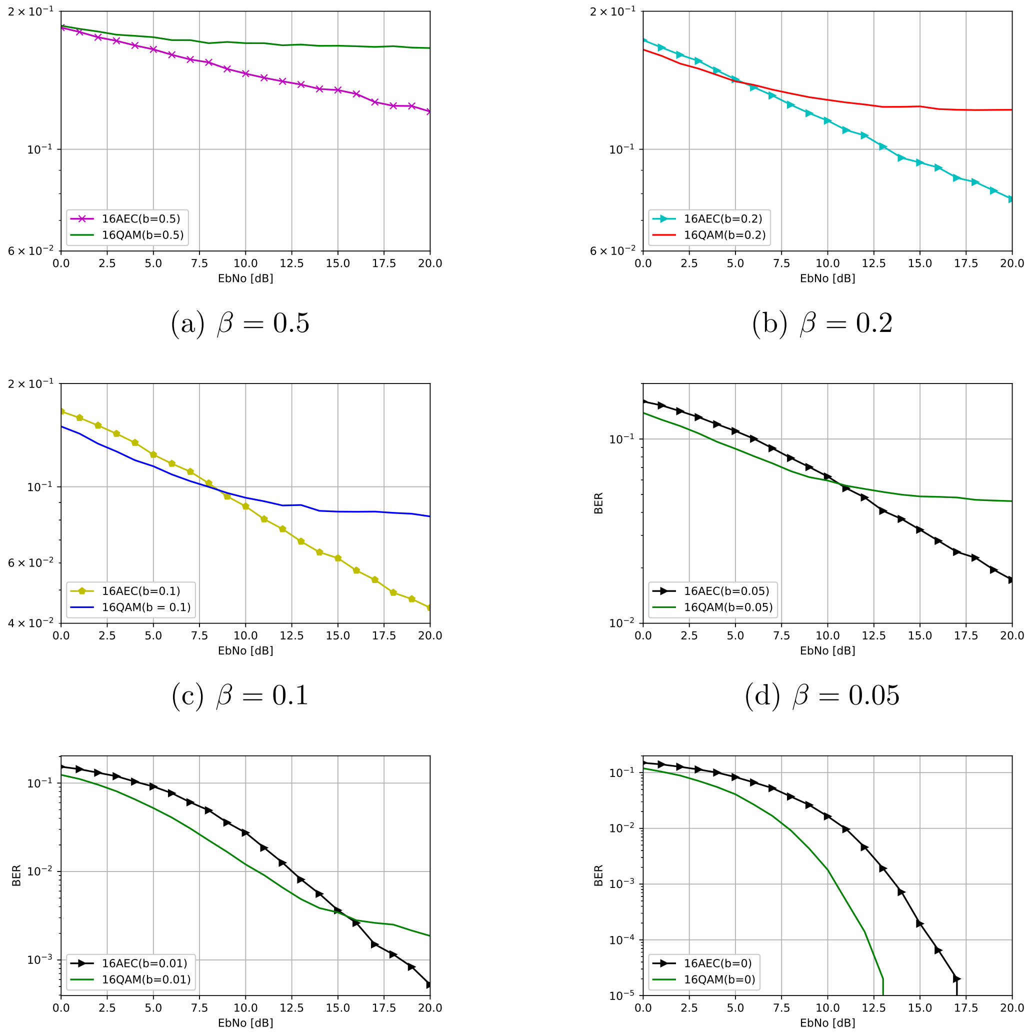
(b) 16QAM($\beta = 0.1$, SNR = 5dB)

Figure 10: Learned and Standard Constellation for 16QAM under different noises.

From Figure 10, we observe that the learned constellation are spread non-linearly. The inner points are packed together near origin while outer points are spread much further. Since points further from origin have higher power, they are pushed even further to mitigate the effect of non-linear noise. While in standard constellation, even though power of the outer points is higher, the constellation are fixed hence suffer from lower performance as compared to learned constellation. Comparison of learned and standard constellation under different noises can be found in Appendix II. From this we can draw a conclusion that, for non-linear noise, the constellation should be non-uniformly distributed. Constellation with higher power should be spread further from origin than constellation with lower power. The performance of the learned constellation against standard one is compared with the help of BER plot in the following section.

4.3 Bit Error Rate for 16QAM

Then BER of the learned constellation and standard constellation was compared for different values of β.



(a) $\beta = 0.5$ (b) $\beta = 0.2$

(c) $\beta = 0.1$ (d) $\beta = 0.05$

(e) $\beta = 0.01$ (f) $\beta = 0$

Figure 11: BER comparison of learned and standard 16QAM.

From the plots for Figure 11, we can see that when the non-linearity is high, the learned constellation out performs the standard one. But as the non-linearity decreases the learned constellation and standard constellation act similar for low SNR while learned one out performs the conventional one at high SNR due to their non-linear distribution of constellation. This trend continues for low β. But finally, learned constellation fails at linear case (β=0) to beat conventional one as the learned constellation are arranged in a circular pattern like 16PSK and also matches the BER performance to 16PSK rather than 16QAM. This might be due to the fact that the NN have been trained for non-linearity. Hence, learned constellation outperformed conventional 16QAM when subjected to different level of non-linearity in noise.

5 Conclusions

This section provides the discussion on results and an overall summary of the research along with the possibilities of extension of the research in the future.

5.1 Summary

This thesis gives a general idea on how a simple neural network in the form of Autoencoder can be used to represent a simple communications system. A neural network learns different features for different modification of the layers, which in our case are the linear and non-linear Gaussian noises. An Autoencoder was able to learn the constellation which have similar performance to that of conventional modulation schemes like QAM, PAM and PSK under regular AWGN channel. With the introduction of non-linear disturbances in the channel, learned constellation from Autoencoder outperformed the conventional constellation. We also showed how results obtained from Autoencoder agreed with the theoretical analysis of 4PAM subjected to non-linear noise. The Autoencoder proposed in this research is simple neural network which shows, with the help of deep learning, communications system can be improved to be resilient of non-linearity due to hardware or even channel imperfections.

5.2 Future Work

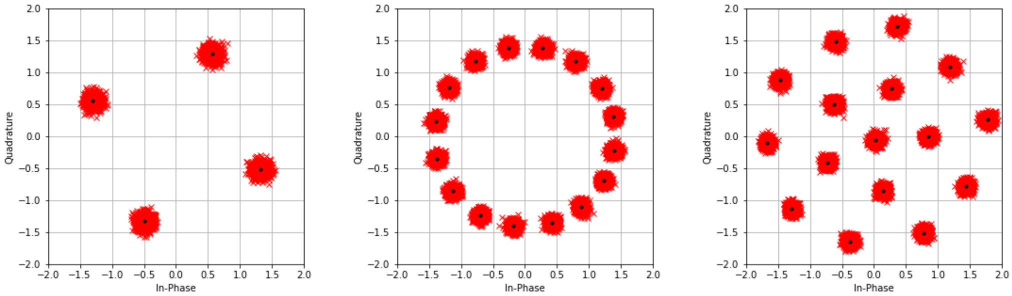
This research involves the use of Autoencoder that was able to represent a simple communications system with only 7 layers. Due to lack of high complexity, it possesses certain drawback which opens up many possibilities of improvement which are discussed below. In daily practice, higher-order schemes like 256QAM, 1024QAM are used. The limitation of our neural network can be seen from its inability to learn the constellation for higher-order modulation. To overcome this problem, we can introduce more complex neural network like Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) which have many hidden layers with very large set of parameters that are learned during training. For instance, we can combine denoising Autoencoder with Convolutional layers which might be able to represent higher-order modulation and even improve the performance of the existing Autoencoder in this paper.

Another possible extension of this research could be the representation of Multi-user communications system. Our Autoencoder is designed to replicate a single user system. Similarly, the Autoencoder can be modified to learn constellation when there are more than one user. This may be possible with introduction of multiple transmitters and receivers who share a common interference channel. It can also be used to represent a MIMO (Multiple Input Multiple Output) channels.

We can also modify the neural network to work with the real-time signal. The neural networks like RNN, Echo-State Network (ESN) and Long Short-Term Memory (LSTM) are able to process real-time signals with the use of buffers and might be able to improve the performance of communications system significantly. We can even add layers for encryption/decryption, power amplifications and soon to represent a real-life communications system. In such cases, Autoencoder can be used for recognition of system parameters like channel or source coding schemes.

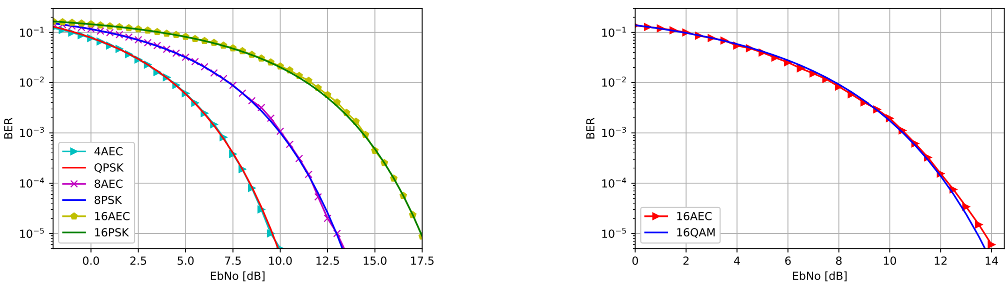
Appendix I

The neural network was also trained for linear Gaussian noise. Some of the constellation obtained and BER comparison plots are given in Figure 12. The Figure 12 shows the constellation and effect of noise (AWGN with 20dB) around them. For obtaining PSK shaped constellation, average energy constraint was applied as it forces all the points to be in a circle. We get QPSK constellation rotated about 45^o. Then, we get slightly rotated constellation for 16PSK. For 16QAM, we see pentagon-shaped constellation. This is obtained by applying average power constraint along with average energy in the normalization layer. This does not look like conventional 16QAM constellation but its BER is comparable with conventional one as shown in Figure 13.



(a) QPSK (b) 16PSK (c) 16QAM

Figure 12: Learned Constellation for different modulation schemes.

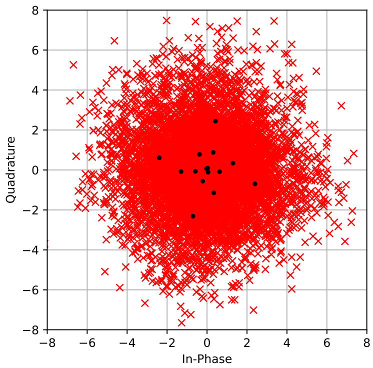


(a) PSK (b) 16QAM

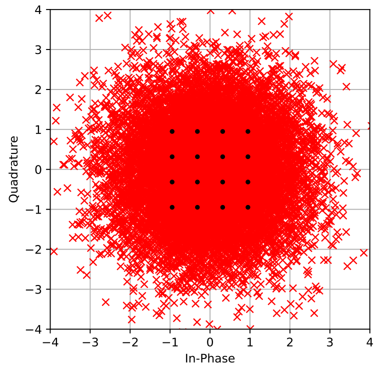
Figure 13: BER comparison of Learned and Standard constellation for different modulation schemes.

Appendix II

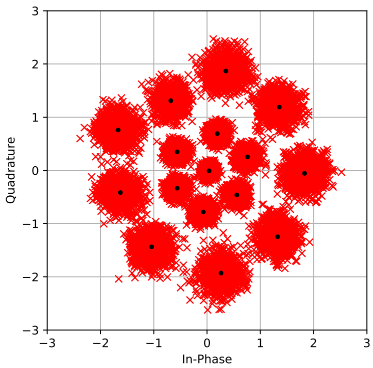
Some of the constellation plots obtained for 16QAM from neural network are shown in Figure 14.



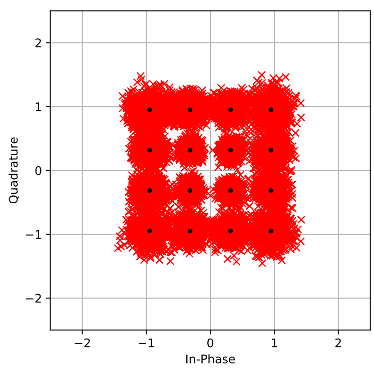
(a) 16AEC ($\beta = 0.5$, SNR $= 5$dB)



(b) 16QAM ($\beta = 0.5$, SNR $=$ 5dB)



(c) 16AEC ($\beta = 0.01$, SNR $=$ 20dB)



(d) 16QAM($\beta = 0.01$, SNR $=$ 20dB)

Figure 14: Learned and Standard Constellation for 16QAM under different noises.

References

[1] B. Aazhang, B. -. Paris, and G. C. Orsak. "Neural networks for multiuser detection in code-division multiple-access communications". In: *IEEE Transactions on Communications* 40.7 (1992), pp. 1212–1222.

[2] S. C. A. Thomopoulos, L. Zhang, and C. D. Wann. "Neural network implementation of the shortest path algorithm for traffic routing in communication networks". In: *[Proceedings] 1991 IEEE International Joint Conference on Neural Networks*. 1991, 2693–2702 vol.3.

[3] L. Shukla. *Designing Your Neural Networks*. https://towardsdatascience.com/designing-your-neural-networks-a5e4617027ed. 2019.

[4] H. Zhou et al. "Research on fiber nonlinear coefficient". In: *2017 16th International Conference on Optical Communications and Networks (ICOCN)*. 2017, pp. 1–2.

[5] Y. C. Gültekin et al. "Constellation shaping for IEEE 802.11". In: *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2017, pp. 1–7.

[6] F. Steiner and G. Boecherer. "Comparison of Geometric and Probabilistic Shaping with Application to ATSC 3.0". In: *SCC 2017; 11th International ITG Conference on Systems, Communications and Coding*. 2017, pp. 1–6.

[7] R. Raich and G. T. Zhou. "On the modeling of memory nonlinear effects of power amplifiers for communication applications". In: *Proceedings of 2002 IEEE 10th Digital Signal Processing Workshop, 2002 and the 2nd Signal Processing Education Workshop*. 2002, pp. 7–10.

[8] H. Iimori and G. T. F. de Abreu. "Rate-optimal communication under nonlinear Gaussian noise via constellation shaping". In: *2018 IEEE Wireless Communications and Networking Conference (WCNC)*. 2018, pp. 1–5.

[9] D. Zhou, K. Niu, and C. Dong. "Constellation shaping for bit-interleaved polar coded-modulation". In: *2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2016, pp. 1–5.

[10] T. O'Shea and J. Hoydis. "An Introduction to Deep Learning for the Physical Layer". In: *IEEE Transactions on Cognitive Communications and Networking* 3.4 (2017), pp. 563–575.

[11] G. T. F. de Abreu. "Supertight algebraic bounds on the Gaussian Q-function". In: *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*. 2009, pp. 948–951.

[12] Wikipedia contributors. *Deep Learning — Wikipedia, The Free Encyclopedia*. [Online; accessed 26-April-2020]. 2020. URL: https://en.wikipedia.org/wiki/Deep_learning.

[13] H. Jaeger. *CO22-320372: Lecture Notes Machine Learning*. http://minds.jacobs-university.de/uploads/teaching/lectureNotes/LN_ML4IMS.pdf. 2019.

[14] P. Baldi. *Autoencoders, Unsupervised Learning, and Deep Architectures*. 2012.

[15] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[16] Wikipedia contributors. *Autoencoder — Wikipedia, The Free Encyclopedia*. [Online; accessed 07-May-2020]. 2020. URL: https://en.wikipedia.org/wiki/Autoencoder.

[17] F. Alberge. "Deep Learning Constellation Design for the AWGN Channel With Additive Radar Interference". In: *IEEE Transactions on Communications* 67.2 (2019), pp. 1413–1423.

[18] A. Goldsmith. *Wireless Communications*. Cambridge University Press, 2005. DOI: 10.1017/CBO9780511841224.

List of Figures

List of Tables