International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

Performance Evaluation of Deep Learning Autoencoder in Single and Multi-Carrier Systems

Srinivasa Rao Kandula
Electronics and Communication Engineering, Dhanekula Institute of Engineering and Technology, Vijayawada, India
ksrinivas.ece@gmail.com

Goda Srinivasa Rao
Computer Science Engineering, KL University , Guntur, India
gsraob4u@gmail.com

Javvaji V K Ratnam
Electronics and Communication Engineering, Kallam Haranatha Reddy institute of Technology, Guntur, India
ratnamjvklakshmi@gmail.com

Kondragunta Rama Krishnaiah,
Computer Science Engineering, R K College of Engineering, Vijayawada, India
kondraguntark@gnail.com

Sharada Mani Dalu
Information technology, QIS college of Engineering and technology, Ongole, India
sharadadalu1234@gmail.com

Koteswararao Seelam
Electronics and Communication Engineering, Kallam Haranatha Reddy institute of Technology, Guntur, India
drkrseelam@gmail.com

Abdul Hussain Sharief
Electronics and Communication Engineering
shariefah@gmail.com

Abstract: An entire system of Single Carrier Communication and Orthogonal Frequency Division Multiplexing is modelled using an autoencoder. The model employs Deep Neural Networks (DNNs) as the transmitter and receiver, responsible for tasks such as encoding, modulation, demodulation, and decoding. The effectiveness of this approach is demonstrated by its ability to outperform traditional communication systems in real-world scenarios that involve channel and interference effects, as measured by the Block Error Rate. AI-enabled wireless systems can overcome limitations of traditional communication systems by learning from wireless spectrum data and optimizing performance for new wireless applications. The aim of this paper is to examine how autoencoder-based deep learning can enhance the performance of a communication system that employs Single Carrier and OFDM. The architecture effectively addresses channel impairments and improves overall performance. The simulation results suggest that even when the autoencoder's channel layer is affected by impairments, autoencoders still outperform traditional communication systems in terms of BLER performance.

Index Terms —Communication systems, Forward Error Correction Coding, OFDM, Deep Neural Network, Autoencoder, BLER.

1. Introduction

Wireless communication has a transformational impact on modern society, but emerging technologies like smart cities and autonomous vehicles present challenges to traditional methods. To overcome these challenges, different architectures, approaches, along with the algorithms are

needed at all stages of the communication system. Conventional communication systems rely on analytical models or expertise in order to construct individual communication blocks for both transmitters and receivers. However, deep learning techniques offers a substitute approach for designing the wireless communication system, including the physical layer. With recent advancements in algorithms and computational capabilities, the communication systems designed based on deep learning technologies can learn and adapt to the available data [1, 2]. Researchers have investigated the applications of ML algorithms in channel coding, decoding, MIMO detection, and communication systems [3-9]. The field of communication encompasses a wealth of specialized knowledge in areas such as information theory, probability, statistics, and mathematical modelling. The field of communication offers several demonstrated techniques and methodologies for optimizing physical layer performance, channel modelling (as described in [10]), and signalling strategies (as described in [11]). A communication system aims to achieve the accurate and reliable transmission of a message, usually in the form of a stream of bits, from its source to its intended destination. This is achieved through the use of a transmitter and receiver, which transmit the information through a channel. In order to optimize performance, it is a standard practice to divide the transmitter and receiver into separate, independent blocks. Each block is assigned a specific function i.e., Source Coding, channel coding, modulation, demodulation, and channel estimation (as explained in [11]). Although dividing the transmitter and receiver into distinct blocks permits customized optimization and regulation of each unit, it may not consistently produce the most optimal results. As reported in [12], there are certain cases in which the block-based approach falls short of optimal performance. However, a communication system based on deep learning optimizes the transmitter and receiver, without the need for separate blocks, following the traditional design of the communication system [12][13]. Capacity-approaching codes were designed in [14] using a novel loss function for AE training. OFDM offers several advantages over conventional single-carrier methods, such as orthogonal sub-carrier frequencies across one OFDM period, enabling digital modulation and multiplexing using IFFT to create required signals. The most significant benefit of OFDM over single-carrier systems is its ability to withstand all channel conditions without time domain equalization, making it less sensitive to sample timing offsets and strong co-channel interference against narrow-bands. Overlapping sub-channels result in efficient spectrum utilization, and flat-fading sub-channels divide the channels, making OFDM more resistant to frequency-selective fading [15]. OFDM uses a modified version of the autoencoder neural network proposed in [16].

In this study, an autoencoder based communication system for implementing SC and OFDM has been explored. A deep neural network, encompassing both an encoder and a decoder, is employed within the system to embody the functionality of the transmitter and receiver, correspondingly. This approach replaces separate encoding and decoding modules with an autoencoder, which learns the concealed representation of the data to reconstruct the input data. The proposed design optimizes the communication process between the transmitter and receiver in a joint manner by utilizing a convolutional encoder-decoder that takes into account channel impairments for a single-antenna system. The performance of the AE model is assessed by measuring its block error rate (BLER) on an Additive White Gaussian Noise (AWGN) channel. It has been demonstrated by the simulation results that the AE-based model we proposed exhibits a BLER that is similar to traditional models that use modulation techniques like BPSK and 16PSK. Additionally, the study shows that the suggested model has better BLER compared to previous studies (references 12, 17, and 18). Our study's findings emphasize the potential of AE-based communication systems to function as an alternative to conventional block-based wireless communication systems.

The structure of this paper is as follows. Section 2 offers a review of relevant literature. Section 3 provides a brief introduction to the AE-based communication system and examines regularization. Section 4 presents the model proposed in this study, while Section 5 covers the simulations and evaluation of the AE system's performance when applied to SC and OFDM. Finally, Section 6 presents a summary of the study's findings and conclusions.

Related works

T. O'Shea et al. introduced the idea of employing autoencoders (AEs) in communication systems, as stated in their works [12] and [13]. A previous work [12] conceptualized the communication system from the perspective of AE, and proposed a method to design the reconstruction task for the system from beginning till end. This method involved simultaneously optimizing the components of both the transmitter and receiver using a feedforward neural network. Another study [13] explored the use of an AE channel for developing complete radio communication systems. In this study, authors approached the learning task as unsupervised machine learning problem and aimed to improve the reconstruction loss by introducing synthetic impairment layers.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

These layers incorporate several regularization techniques that replicate the common impairments encountered in wireless channels. Additionally, [20] examines an optical wireless communication system that serves a single user, utilizing AEs. In [22], for end-to-end learning the authors proposed an extended channel AE model, which is particularly useful when the channel response characteristics are uncertain or challenging to model. Adversarial approach was used to approximate the channel response and encode information, allowing both tasks to be learned as one over a varied channel condition. Authors have demonstrated the effectiveness of this model in an over-the-air system through training and validation. Another study [23] investigated the optimizers' impact on the speed of convergence pertaining to AE applications when channel exhibits the properties of both high-mobility as well as of short-coherence. Furthermore, end-to-end learning has demonstrated potential in intricate communication scenarios for example in molecular as well as in optical communications, as evidenced by the encouraging performance shown in previous studies [24, 25]. Furthermore, we assess various channel uses and modulation techniques in our design and the constellations produced by the autoencoder. The findings highlight the effectiveness of optimizing with deep learning techniques in creating innovative methods for wireless communication design.

2. Wireless communications making use of Channel autoencoder

2.1. Traditional Wireless Communication System

Typically, there are three essential components i.e., a transmitter, a receiver, and a channel in the wireless communication system. The message s is sent by the transmitter to the receiver, that is chosen from a set of M possible messages where s ∈ M = {1, ..., M}, by utilizing the channel n times. Message s when undergoes digital modulation f: $M \mapsto R_n$, results in a vector $x = f(s) \in R^n$ that is transmitted. The process of modulation involves mapping the discrete symbols of the input alphabet to complex numbers, which in turn correspond to points on the constellation diagram. The transmitted signal's power is controlled by imposing power restrictions on x by the transmitter which can take the form of a constraint on energy such as $\|x\|_2^2 \leq n$ or a mean power constraint like $E[|x_i|^2] \leq 1$ for all i. With $k = log_2(M)$ bits representing each message s, the communication system works at rate R, which is k/n, where R is calculated in bits per channel use. Whenever, the transmitted symbols pass through the channel, they are subject to distortions. When the receiver receives these symbols, it produces the estimate ŝ of message s that was originally transmitted. We can define Block Error Rate (BLER) *Pe* as probability that the estimate ŝ produced by the receiver does not match the originally transmitted message s, which can be expressed as (1):

$$P_e = \frac{1}{M}\sum_s P_r(\hat{s} \neq s/s) \quad (1)$$

Multiple independent blocks are involved in a conventional communication system, the way shown in Figure 1. The source encoder in a communication system is tasked with compressing input data and removing redundancies, post which channel encoder adds controlled redundancy to the compressed data. Forward error correction, which is also referred to as channel coding, is a commonly used technique in wireless communication systems. Its primary purpose is to ensure that the received data is identical to the transmitted data, particularly since wireless links are susceptible to interference and fading, both of which can result in errors. In order to address this issue, the transmitter employs a technique called coding, which involves appending additional information to the data prior to transmission. Coding helps to reduce the negative impact of the communication medium on the transmitted data. In contrast, an uncoded communication system does not involve the inclusion of additional information to conceal the data being transmitted. If the modulation technique employed by the transmitter is flexible, the modulator block modifies the signal's features according to the chosen data rate along with the signal level at the receiver. As the transmitted signal travels through the channel, it undergoes both distortion and weakening. Subsequently, when the signal reaches the receiver, additional noise is introduced as a result of hardware impairments. The transmitter designs each communication block to enhance system efficiency and prepare the signal to withstand unfavourable effects of the communication channel and also of inherent receiver noise. Receiver undertakes analogous operations in the opposite order to reassemble the information that was originally transmitted.

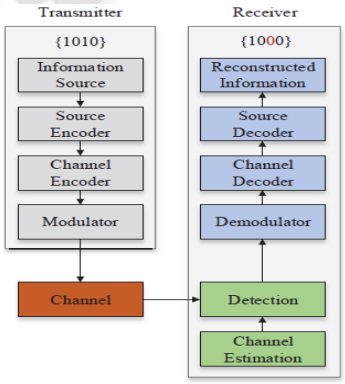

Fig. 1. Typical wireless communication system showcasing blocks of channel coding & modulation

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

2.2. An Optimization Process with Autoencoder

An autoencoder (AE) is a kind of Feedforward Neural Network (FNN) in which input and output layers are the same. The initial autoencoder is unsupervised deep learning algorithm that compresses the inputs for acquiring a condensed representation. The reduced representation generated by the AE is utilized to reconstruct initial inputs at output layer, resulting into a compressed representation of the original input [26]. An autoencoder is comprised of hidden layer, which serves to represent the input code. In general, an autoencoder (AE) is composed of two primary components i.e., an encoder f(s) that compresses the input s into a compact form y, and a decoder g(y) which reconstructs the original input r from the compressed form y [26]. The AE typically consists of one or more hidden layers, in which every layer is defined by bias vectors b and also by weight matrices W. Mos simple form of AE has only single hidden layer.

$$y = f(x) = s_1\left(W^{(1)}x + b^{(1)}\right), \tag{2}$$

$$r = g(y) = s_2\left(W^{(2)}y + b^{(2)}\right), \tag{3}$$

The activation functions s_1 and s_2, which are usually non-linear, are denoted in the equation.

2.3. Autoencoder System for Single Carrier Communication

The communication system is conceptualized as an autoencoder that aims to minimize error when reconstructing the transmitted messages at the receiver. It is possible to regard the decoder and encoder as fulfilling the roles of the receiver and transmitter respectively. A standard AE architecture for learning of a complete communication system, as suggested by [12], is illustrated in Figure 2. The architecture depicts the transmitter as a FNN to impose physical constraints pertaining to transmit vector x, that includes dense layers and normalization layer.

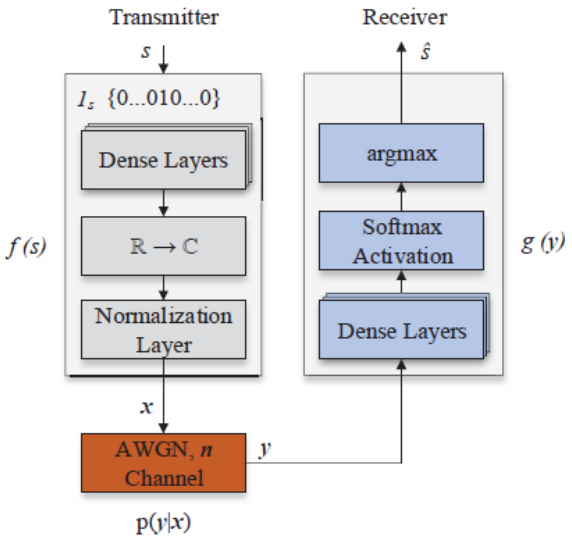

Fig. 2. Autoencoder representation of structure of wireless communication system.

As previously discussed, a common communication system comprises different blocks that carry out tasks like channel encoding, decoding, modulation, and demodulation. On the other hand, the Autoencoder based system does not consist of discrete blocks, but instead aims to optimize the system as a whole through an end-to-end process that incorporates parameters of the system including size of i/p, number of channels used for every message, and the constraints of transmit signal power. The autoencoder models are implemented based on these parameters, which resemble those used in traditional communication systems with channel coding or uncoded communication. The performance of these models is then evaluated over an AWGN channel for comparison. The input can be represented by s, channel by cha(s), encoder layer by enc(s), and decoder layer represented by dec(s). The AE is trained using Adam (**Ada**ptive **m**oment estimation) optimizer to produce an output $dec(cha(enc(s)))$ that minimizes an arbitrary loss function L (s, $dec(cha(enc(s)))$) [27].

The autoencoder using only one antenna generates constellation diagrams that are not fixed in advance. Rather, they are learned in accordance with the specific receiver performance metric targeted for minimization, such as symbol error rate, coherence time including propagation loss. Constraints outlined in reference [28] are enforced by the transmitter's hardware as follows:

(a) For Energy $\|x\|_2^2 \leq$ n,
(b) For Amplitude $|x_i| \leq 1\ \forall_i$,
(c) For Average power E $[|x_i|^2] \leq 1\ \forall_i$ on x.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

System's data rate can be determined by the equation $R = k/n$ which is bit per channel use. The parameter k denotes input bits and is equal to $log_2(M)$, where M denotes the number of feasible messages those are transmitted. The parameter n encompasses not only the input bits but also additional bits employed to mitigate the effects of channel. The (n, k) representation denotes that the system transmits single message out of M possible messages (characterized by k bits) through n channel uses. Block diagram of channel autoencoder is depicted in Figure 2, which utilizes distribution of the channel data to learn and compensate for impairments. The channel in a communication system is categorized by the conditional PDF $p(y|x)$, where y belongs to the n-dimensional space R^n and represents received signal. Received signal at receiver side is denoted as y, followed by operation r: $R^n \rightarrow M$, that is performed to evaluate the transmitted message s. Channel autoencoder aims to optimize the mapping from x to y, enabling the recovery of s by minimizing the probability of error.

2.4. Autoencoder System for OFDM

The Fig. 3 shows a OFDM autoencoder communications system. The transmitter's encoder first maps each set of k information bits in a sequence to message s in a way that $s \in \{0, \ldots, M-1\}$, wherein $M = 2^k$, to form T messages. Each of the T messages, s, is transformed to n real-valued channel uses using an encoder function $f(s) = \mathbf{x}$, where $x \in \mathbb{R}^n$. Outcome of this is an actual code rate R indicating k/n bits per physical channel. The information can be rewritten as: "Next, the encoded message $x = f(s) \in \mathbb{R}^n$ is mapped into a complex symbol $\mathbf{x}_C = g(s) \in \mathbb{C}^{n/2}$, where every two real channel uses correspond to one complex symbol. The normalization layer in the encoder imposes constraints on the input signal x, which results in further restrictions on the encoded symbols. To demonstrate the potential constraints, the normalization layer is utilized:

- Constraint on Energy: $\|x\|_2^2 \leq n$,
- Constraint on Average power: $E[|x_i|^2] \leq 1, \forall i$

Normalized symbols are mapped onto the OFDM subcarriers and passed through an AWGN channel.

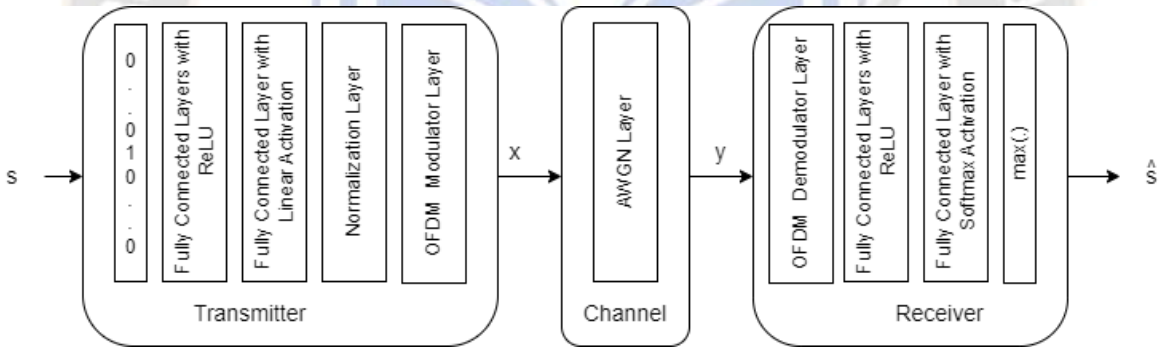

Fig. 3. Autoencoder representation of an OFDM system's structure.

The transmitter encodes s and outputs encoded symbols, x. The channel impairs the encoded symbols to generate $\mathbf{y} \in \mathbb{R}^{n/2}$. The transmitted message s is decoded by the receiver to output an estimate s̃. Input message is represented as one one-hot vector $\mathbf{1}_S \in \mathbb{R}^M$, where all elements are having zero values except for the s^{th} one. AWGN channel introduces noise for achieving the desired SNR.

Training the neural network at several SNR levels ensures that the autoencoder can handle a range of SNR values without retraining. Set training SNR values as an array between -1dB and 9dB. Generate multiple batches of training sequences, where each batch (cell) experiences a different SNR. Set random number generator state for repeatable results for demonstration purposes only.

The key components of the autoencoder are as follows: -

a) *Input*: Here, symbol s is transformed into one-hot vector, meaning that it can only have valid combinations of values where one bit is set to '1' and all the others are set to '0'. This encoding scheme allows a state machine to operate at a higher clock rate compared to other encoding methods. Moreover, defining the state of a one-hot vector requires accessing only one flip-flop, which has a low and consistent cost.

b) *Transmitter*: In this component, transmitter consists of a FNN having many dense layers and the output of final dense layer is adapted to represent two complex numbers for every input symbol post modulation. The two complex numbers generated by the last dense layer of the transmitter correspond to the in-phase real part I and quadrature imaginary part Q. A normalization layer

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

is included in the transmitter to ensure that the encoding symbols satisfy the physical constraints imposed on x.

c) *Channel*: Channel layer is not subjected to training and is presumed to be fixed. Channel layer is characterized by AWGN layer, which is a fixed component added to the signal. The parameter β, which determines the variance of the AWGN added to the signal, is calculated as $\beta = (\frac{2RE_b}{N_0})^{-1}$, Here, the ratio of Bit Energy (E_b) to Noise Power Spectral Density (N_0) is used to calculate β. For each training example, the value of β varies and is used to determine the noise variance that is added to the signal in the forward pass to simulate signal distortion. However, during the backward pass, the noise is not taken into consideration.

d) *Receiver*: similar to transmitter, it is constructed using a Fully connected Neural Network (FNN). Its final layer employs the softmax activation function to generate a probability vector $p \in (0, 1)^M$ representing all potential messages. Value in p with the greatest probability is designated as $\hat{s}$.

e) *Training*: Here, training is imparted to autoencoder through the Adaptive Moment (Adam) optimizer to adjust the weights of the FNN, and its performance is then evaluated. The training batch comprises all potential messages $s \in M$.

3. Results and Evaluation of Simulation Performance

The autoencoder functions by using the data that is generated during transmission along with the corresponding data present at the receiving side. Since this data utilized by the autoencoder is not labelled externally, it is categorized as an unsupervised learning system. This approach enables the autoencoder to acquire knowledge without any prior information. A vector with only single element being "1" and others being "0 represents input message. This is known as a one-hot vector. The channel through which the message passes is AWGN channel. For attaining the particular bit energy to noise power ratio, the AWGN channel introduces noise to the message. A (7,4) autoencoder network was presented by researchers in reference [29], which employs energy normalization and is trained at 3 dB. For optimal outcomes with minimal complexity, the encoder (transmitter) and decoder (receiver) consisted of two fully connected layers and the input layer accepts a one-hot vector of length M. Encoder's initial fully connected layer has M number of inputs and outputs, which are followed by Rectified Linear Unit (ReLU) layer. A one-hot vector having a length of M is fed into the input layer. The encoder's first fully connected layer comprises M number of inputs and outputs, which are then succeeded by a Rectified Linear Unit layer. Subsequent to the M number of inputs of second fully connected layer, there are n numbers of outputs, which are succeeded by a layer performing normalization. Implementation of AWGN channel layer has been undertaken after the encoder layers. The output of channel is subsequently fed into the layers undertaking decoding, beginning with fully connected layer containing n number of inputs, M outputs and succeeded by ReLU layer. In next fully connected layer, there are M numbers of inputs and outputs, which are succeeded by the softmax layer that generates likelihood of every M symbol. Ultimately, transmitted symbol which is most probable ranging from 0 to M-1 is identified by the classification layer.

A (2,2) autoencoder is trained using below specific parameters, including energy normalization.

- Optimizer is based on Adaptive moment estimation
- 0.01 is initial learning
- 15 epochs are the maximum limit
- 20*M is the size of minibatch
- Drop period and drop factor of 10 and 0.1 respectively are set for schedule of piecewise learning

Adaptive Moment Estimation optimizer algorithm can be used to train the autoencoder's parameters to optimally reduce the reconstruction error of received signal. The Adam optimizer algorithm uses a combination of momentum and adaptive learning rate to converge efficiently to the lowest value of the cost function. When it comes to an autoencoder, cost function is determined by the reconstruction error observed between input and output signals.

During training, the Adam optimizer algorithm updates the parameters of the autoencoder using the following steps:

1. Initialize the first and second moment vectors as $m_0 = 0$ and $m_0 = 0$ respectively.
2. Forward pass through the autoencoder to obtain the output signal.
3. Calculate the reconstruction error from the values of input and output signal.
4. Calculate the cost function's gradient $g_t = \nabla_\theta J(\theta_t)$
5. Update the first moment estimate
   $m_t = \beta_1 * m_{t-1} + (1 - \beta_1) * g_t$
   where β_1 is the first moment estimate's exponential decay rate, typically 0.9.
6. Update the second moment estimate:
   $v_t = \beta_2 * v_{t-1} + (1 - \beta_2) * g_t^2$
   where β_2 is the second moment estimate's exponential decay rate, typically 0.999.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

7. Compute the first and second moment estimate $\hat{m}_t$ and $\hat{v}_t$post bias-correction using below-mentioned equations:

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

where t is the current iteration step, for first moment estimate's exponential decay rate is represented by β_1, while that for the second moment estimate being represented by β_2.

In addition to the standard hyperparameters used in the Adam optimizer, such as the learning rate and first and second moment estimates' exponential decay rates, there are additional hyperparameters that are specific to the application of autoencoders for wireless communication. These include the Signal-to-Noise Ratio and the batch size.

Fig. 4 shows the training process at 3dB noise level. The validation accuracy rapidly exceeds 90%, whereas the validation loss consistently reduces. This suggests that the initial training value was appropriately set to allow some errors but still achieve convergence.

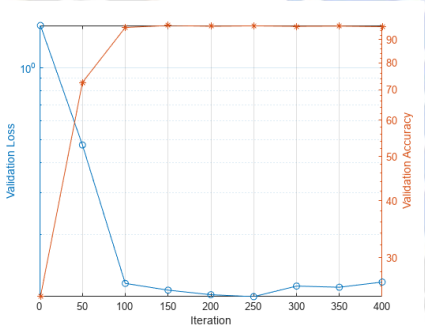

Fig. 4. The training process plot.

Fig.5 displays the layer diagrams of the complete autoencoder, including its encoder and decoder networks, represented by the objects generated from the trained network. The encoder network is also known as the transmitter, while the decoder network is known as the receiver.

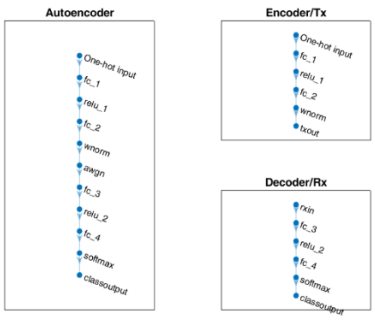

Fig. 5. The objects generated by the trained network.

The (2,2) autoencoder's Block Error Rate (BLER) performance was tested by simulating the encoding of random information bits into complex symbols. The symbols were passed through an AWGN channel to simulate impairment and then decoded. The simulation was run for at least 10 block errors and compared to an uncoded QPSK system with block length 2.

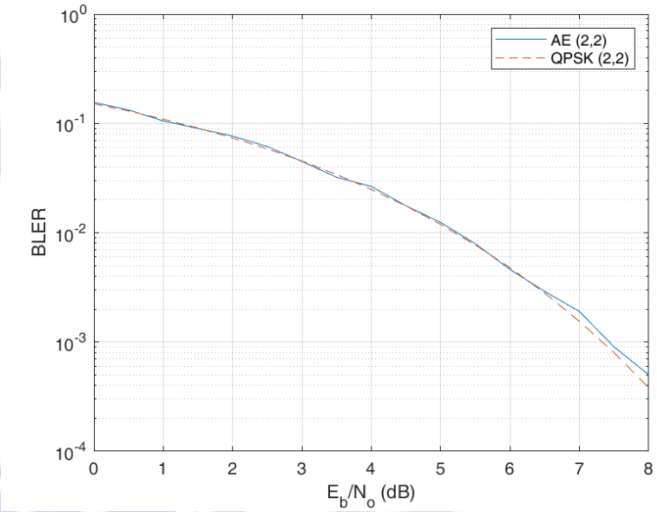

Fig. 6. The BLER plot for QPSK (2,2) and AE (2,2).

The (2,2) autoencoder has been trained to attain a convergence point on a QPSK constellation with an optimal phase shift for prevailing channel states. Conversely, (2,4) autoencoder converges on a constellation map of 16-PSK having a phase-shift when energy normalization is utilized. It's worth emphasizing that energy normalization is employed to guarantee that each symbol has equivalent energy and is positioned on the unit circle. Subject to this limitation, the most optimal constellation is the PSK constellation with symbols equally spaced at angular intervals. In conclusion, (2,4) autoencoder converges on three-tier constellation that includes symbols 1-6-9 while utilizing mean normalization of power. A comparison simulation was carried out to evaluate the performance of the (7,4) autoencoder in terms of BLER against that of the (7,4) Hamming code implemented along with QPSK modulation, using both types of decoding i.e., hard decision and maximum likelihood. The baseline utilized was an uncoded (4,4) QPSK system, which essentially functioned as a PSK modulated system which transmitted blocks comprising of 4 bits and measured BLER.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

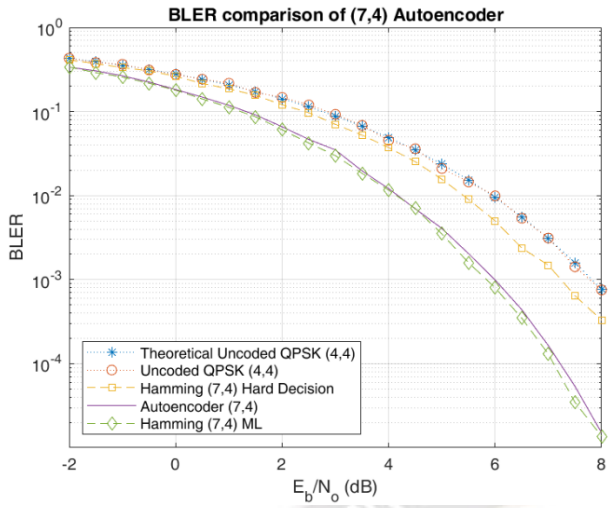

Fig. 7. (7, 4) Autoencoder BLER performance comparison.

Subsequently, simulations were carried out to compare the block error rate performance of autoencoders having code rate R=1 with the QPSK systems, which is uncoded. To compare and evaluate the BLER performance, simulations were carried out on (2,2) and (8,8) uncoded QPSK systems, which were taken as the baselines. Furthermore, these systems' performance was compared against (2,2), (4,4), and (8,8) autoencoders.

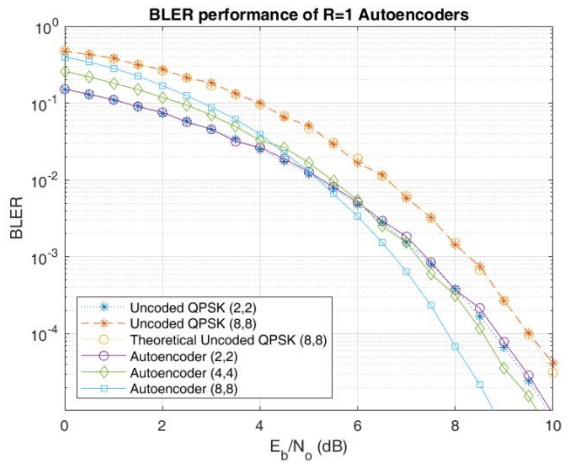

Fig. 8. Autoencoder of R = 1 BLER performance comparison.

The QPSK bit error rate was found to be identical for both the (8,8) and (2,2) cases. However, it was observed that the block length n had an impact on the BLER, which worsened as n increased following relationship $BLER = 1 - (1 - BER)^n$. The observed BLER performance of the (8,8) QPSK system was found to be worse than that of the (2,2) QPSK system, as anticipated. The study found that the BLER performance of the (2,2) autoencoder was similar to that of (2,2) QPSK. However, for (4,4) and (8,8) autoencoders, a coding gain was achieved by jointly optimizing the channel coder and constellation, leading to better BLER performance compared to their corresponding uncoded QPSK systems.

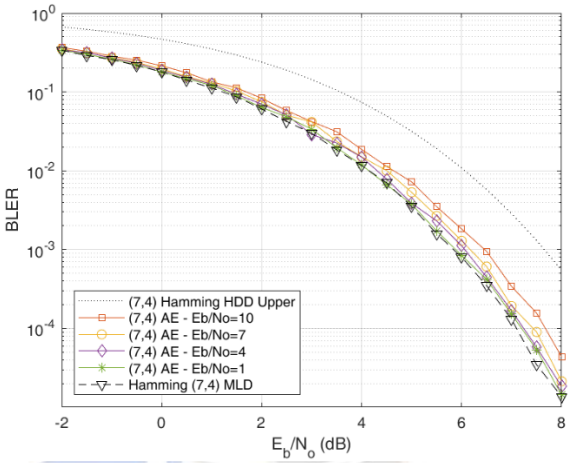

Fig. 9 Autoencoder with Hamming code performance comparison.

After training with energy normalization under various values, (7,4) autoencoder's BLER performance was compared. Performance of BLER for (7,4) autoencoder having energy normalization was plotted in Figure 10, along with the simulated BLER of Maximum Likelihood Decoded (MLD) and the upper bound in respect of hard decision decoded Hamming (7,4) code calculated theoretically. As the Eb/N0 decreased from 10 dB to 1 dB, BLER performance of (7,4) autoencoder's was seen to approach the same as (7,4) Hamming code with MLD, which nearly matched the MLD (7,4) Hamming code at that point. This finding is important as it highlights the capability of autoencoders to autonomously learn combined coding and modulation strategies.

The OFDM system was implemented by setting the number of subcarriers, Nfft to 256. The two fully connected layers mapped k bits (in the form of length M one-hot arrays) into n real numbers, resulting in a rate R=k/n communications system. Once normalized, the OFDM modulator layer allocated each of the n real numbers to n/2 complex-valued symbols and assigned each symbol to a subcarrier. To ensure that the OFDM modulator layer outputted full OFDM symbols, the minimum input length of the sequence input layer in the third dimension (T) was set to N_{fft}. Therefore, the input to the neural network was a sequence of one-hot values with size $M \times N_{fft}$.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Fig.10 displays the layer diagrams of the complete autoencoder of OFDM system, including its encoder and decoder networks, represented by the objects generated from the trained network.

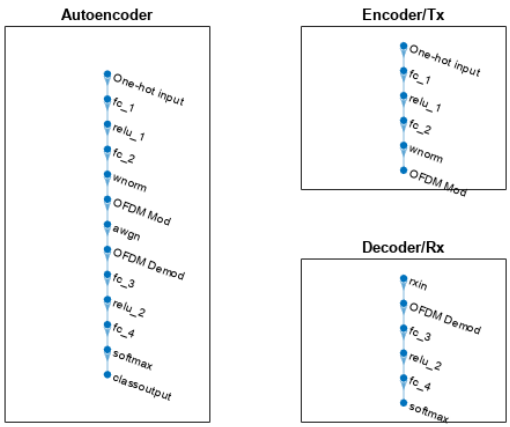

Fig. 10. The objects generated by the trained network.

To simulate an OFDM system, the initial step involves generating k random information bits represented as integers in the range [0, M-1]. These bits are then transformed into complex symbols. Real valued vector x is transformed into a complex valued vector x_c by autoencoder, with the even and odd elements of x being allocated to the quadrature and in-phase components of complex symbol respectively. Specifically, x_c = x(1:2:end) + jx(2:2:end), where j is the imaginary unit. After encoding, complex symbols are passed through the channel, which is AWGN. Complex symbols, impaired by channel, subsequently decoded using autoencoder. MATLAB code runs the simulation for each SNR, with a minimum of 100 block errors or a maximum of 2000 frames per point. A comparison is made between the performance of the autoencoder and uncoded QPSK (uncoded) system with a block length n of 6. In present case, autoencoder can provide a greater coding gain than a basic repetition code. Additionally, it provides approximately 5.5 dB gain compared to an uncoded QPSK system with the same block length in Fig. 11

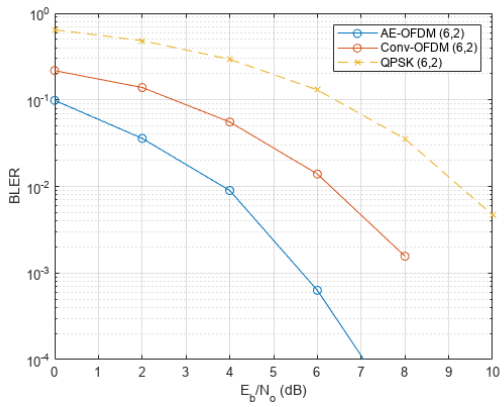

Fig. 11 Autoencoder with (6, 2) code performance comparison.

The Block Error Rate (BLER) results in Fig. 11 indicate that incorporating expert knowledge in the form of OFDM modulation and demodulation into the neural network enables training of an OFDM-based autoencoder. By allowing for multiple channels uses per input symbol (n>k), the autoencoder can learn to achieve better coding gain compared to simple repetition codes.

4. Conclusion and Future scope

In this paper, use of deep learning architectures to enhance performance of single carrier and OFDM communication systems was explored. The researchers employed an autoencoder as a unified transmitter/ receiver for undertaking communication, enabling the optimization of complete communication system, as described in this study. Conventional approach of optimizing individual blocks was replaced with a holistic approach to minimize reconstruction loss. The authors demonstrated that this approach is effective in capturing channel impairments and matching modulation techniques using off-the-shelf DNNs. The paper concludes that Autoencoders can design the complete communication system in an unsupervised way by learning 'coding and modulation' as a single entity. Future research can focus on exploring various learning strategies, including weight initialization, hyperparameter selection, and different emerging autoencoder architectures. This methodology can be further extended to various other communication systems such as multiuser and multiple-antenna systems by incorporating additional autoencoders. The application can also be extended to 5G MIMO systems, Backhaul radios, Satellite Communications etc among others.

References

[1] E.A.A. Alaoui, S.C.K. Tekouabou, S. Hartini, Z. Rustam, H. Silkan, S. Agoujil, "Improvement in automated diagnosis of soft tissues tumors using machine learning", Big Data Min. Anal. 4 (1) (2021) 33–46, http://dx.doi.org/10.26599/ BDMA.2020.9020023.

[2] S. Shorewala, A. Ashfaque, R. Sidharth, U. Verma, "Weed density and distribution estimation for precision agriculture using semi-supervised learning", IEEE Access 9 (2021) 27971–27986, http://dx.doi.org/10.1109/ACCESS.2021.3057912.

[3] M. Varasteh, J. Hoydis, B. Clerckx, "Learning to communicate and energize: Modulation, coding, and multiple access designs for wireless information-power transmission", IEEE Trans. Commun. 68 (11) (2020) 6822–6839, http://dx.doi.org/10.1109/TCOMM.2020.3017020.

[4] J. Ren, G. Yu, G. Ding, "Accelerating DNN training in wireless federated edge learning systems", IEEE J. Sel. Areas Commun. 39 (1) (2021) 219–232, http: //dx.doi.org/10.1109/JSAC.2020.3036971.

[5] M.E. Morocho-Cayamcela, H. Lee, W. Lim, "Machine learning for 5G/B5G mobile and wireless communications: Potential, limitations, and future directions", IEEE Access 7 (2019) 137184–137206.

[6] M.E. Morocho Cayamcela, W. Lim, "Artificial intelligence in 5G technology: A survey", in: 2018 International Conference on Information and Communication Technology Convergence (ICTC), 2018, pp. 860–865.

[7] M.E. Morocho-Cayamcela, H. Lee, W. Lim, "Machine learning to improve multihop searching and extended wireless reachability in V2x", IEEE Commun. Lett.(2020) 1.

[8] J.N. Njoku, M.E. Morocho-Cayamcela, W. Lim, CGDNet: "Efficient hybrid deep learning model for robust automatic modulation recognition", IEEE Netw. Lett. 3 (2) (2021) 47–51, http://dx.doi.org/10.1109/LNET.2021.3057637.

[9] J.N. Njoku, M.E. Morocho-Cayamcela, W. Lim, "Automatic radar waveform recognition using the wigner-ville distribution and AlexNet-SVM", in: Proceedings of the KICS Summer Conference, Pyeongchang, South Korea, 2020, pp. 1–4.

[10] P. Popovski, "A mathematical view on a communication channel, in Wireless Connectivity: An Intuitive and Fundamental Guide", Wiley, Wiley, 2020, pp.145–173, http://dx.doi.org/10.1002/9781119114963.ch6.

[11] F. Farzaneh, A. Fotowat, M. Kamarei, A. Nikoofard, M. Elmi, "The amazing world of wireless systems, in: Introduction to Wireless Communication Circuits", River Publishers, 2020, pp. 1–26.

[12] T. O'Shea, J. Hoydis, "An introduction to deep learning for the physical layer", IEEE Trans. Cogn. Commun. Netw. 3 (4) (2017) 563–575.

[13] T.J. O'Shea, K. Karra, T.C. Clancy, "Learning to communicate: Channel autoencoders, domain specific regularizers, and attention", in: 2016 IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2016, 2017, pp. 1–6.

[14] N. A. Letizia and A. M. Tonello, "Capacity-Approaching Autoencoders for Communications," 2020, available on arXiv:2009.05273.

[15]. Chadha, A., Satam, N., &Ballal, B. (2013). "Orthogonal Frequency Division Multiplexing and its Applications" arXiv preprint arXiv:1309.7334.

[16] A. Felix, S. Cammerer, S. Dörner, J. Hoydis and S. Ten Brink, "OFDM-Autoencoder for End-to-End Learning of Communications Systems," *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2018, pp. 1-5, doi: 10.1109/SPAWC.2018.8445920.

[17] H. Zhang, L. Zhang, Y. Jiang, "Overfitting and underfitting analysis for deep learning based end-to-end communication systems", in: 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP), 2019, pp. 1–6, http://dx.doi.org/10.1109/WCSP.2019.8927876.

[18] L. Liu, T. Lin, Y. Zhou, "A deep learning method-based receiver design", in: 2020 IEEE 6th International Conference on Computer and Communications (ICCC), 2020, pp. 975–979, http://dx.doi.org/10.1109/ICCC51575.2020.9344965.

[19] L. Liu, Y. Luo, X. Shen, M. Sun, B. Li, "β-Dropout: A unified dropout", IEEE Access 7 (2019) 36140–36153, http://dx.doi.org/10.1109/ACCESS.2019.2904881.

[20] M. Soltani, W. Fatnassi, A. Aboutaleb, Z. Rezki, A. Bhuyan, P. Titus, "Autoencoder based optical wireless communications systems", in: 2018 IEEE Globecom Workshops, GC Wkshps 2018 - Proceedings, Institute of Electrical and Electronics Engineers Inc., 2019, pp. 1–6, http://dx.doi.org/10.1109/GLOCOMW.2018.8644104.

[21] T.J. O'Shea, T. Erpek, T.C. Clancy, "Physical layer deep learning of encodings for the MIMO fading channel, in: 55th Annual Allerton Conference on Communication", Control, and Computing, Allerton 2017, 2018-January, Institute of Electrical and Electronics Engineers Inc., 2017, pp. 76–80, http://dx.doi.org/10.
1109/ALLERTON.2017.8262721.

[22] T.J. O'Shea, T. Roy, N. West, B.C. Hilburn, "Physical layer communications system design over-the-air using adversarial networks", in: European Signal Processing Conference, 2018-Septe, 2018, pp. 529–532, http://dx.doi.org/10. 23919/EUSIPCO.2018.8553233.

[23] M.E. Morocho Cayamcela, J.N. Njoku, J. Park, W. Lim, "Learning to communicate with autoencoders: Rethinking wireless systems with deep learning", in: 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC), Fukuoka, Japan, 2020, pp. 308–311.

[24] S. Mohamed, J. Dong, A.R. Junejo, D.C. Zuo, "Model-based: End-to-end molecular communication system through deep reinforcement learning auto encoder", IEEE Access 7 (2019) 70279–70286.

[25] H. Lee, S.H. Lee, T.Q.S. Quek, I. Lee, "Deep learning framework for wireless systems: Applications to optical wireless communications", IEEE Commun. Mag. 57 (3) (2019) 35–41.

[26] D. Wu, M. Nekovee, Y. Wang, "Deep learning-based autoencoder for m-user wireless interference channel physical layer design", IEEE Access 8 (2020) 174679–174691, http://dx.doi.org/10.1109/ACCESS.2020.3025597.

[27] D.J. Ji, J. Park, D.-H. Cho, ConvAE: "A new channel autoencoder based on convolutional layers and residual connections", IEEE Commun. Lett. 23 (10) (2019) 1769–1772, http://dx.doi.org/10.1109/lcomm.2019.2930287.

[28] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," in IEEE Transactions on Cognitive Communications and Networking, vol. 3, no. 4, pp. 563-575, Dec. 2017, doi: 10.1109/TCCN.2017.275837.

[29] T. J. O'Shea, T. Erpek, and T. C. Clancy, "Physical layer deep learning of encodings for the MIMO fading channel," in 2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton). IEEE, 10 2017, pp. 76–80.