INTERNSHIP PROJECT REPORT

On

Classification and Estimation of BPSK Modulated Symbols

BY

Kevin Shaju

Final Year

DISCIPLINE OF ELECTRONICS AND COMMUNICATION ENGINEERING

Madhav Institute of Technology and Science

2022

Classification and Estimation of BPSK Modulated Symbols

A PROJECT REPORT

Submitted in fulfillment of the
Requirements for Internship

Of
Machine Learning and Deep Learning

Submitted by:
Kevin Shaju
Guided by:
Dr. Vimal Bhatia

INDIAN INSTITUTE OF TECHNOLOGY INDORE
MAY 2022

Area of Online Internship	Deep Learning in Wireless Communication
Intern Name	Kevin Shaju
Name Of Institution	INDIAN INSTITUTE OF TECHNOLOGY, INDORE
Faculty Mentor Name	Dr. Vimal Bhatia
Duration	4 MONTHS (10/01/2022 TO 10/05/2022)
Date Of Submission	10/05/2022

Table of Contents

CANDIDATE'S DECLARATION

We hereby declare that the project entitled **"Classification and Estimation of BPSK Modulated Symbols"** submitted in fulfillment for internship completed under the supervision of **Dr. Vimal Bhatia** IIT Indore is an authentic work.

Further, I/we declare that I/we have not submitted this work for the award of any other degree elsewhere.

Name of the student(s) with date

Kevin Shaju

10/05/2022

CERTIFICATE by BTP Guide(s)

It is certified that the above statement made by the students is correct to the best of my/our knowledge.

Signature of BTP Guide(s) with dates and their designation

PREFACE

This report on “Classification and Estimation of BPSK Modulated Symbols” is prepared under the guidance of Dr. Vimal Bhatia. The purpose of classification and estimation of BPSK modulated symbols is to denoise the recieved signal which is transmitted over a wireless channel using Denoising Autoencoders (DAE). The autoencoder is built using Deep learning algorithms i.e., Convolutional Neural Network (CNN) to execute feature extraction and classification. This method achieves impressive results on simulated data but needs to be tested on real world data.

Kevin Shaju
B.Tech. IV Year
Discipline of Electronics and Communication
Madhav Institute of Technology and Science

Gwalior (MP)

ACKNOWLEDGEMENTS

I wish to thank Dr. Vimal Bhatia for his kind support and valuable guidance, who gave me this opportunity to work on this project. I got to learn a lot from this project about Machine learning and Deep Learning.

It is his help and support, due to which I became able to complete the report and Project code.

Without his support this report would not have been possible.

Kevin Shaju
B.Tech. IV Year
Discipline of Electronics and Communication
Madhav Institute of Technology and Science

Gwalior (MP)

ABSTRACT

Denoising of signals transmitted over a wireless channel is a key process in wireless communication. In this report we explore a Deep Learning based solution for this task. First, we study several denoising methodologies in Deep Learning for denoising of data. We then test the Denoising Autoencoder (DAE) on the data which is simulated to be transmitted over a channel.

Preprocessing and scaling are done to make the training more robust and the network able to generate more accurate results. The purpose of this thesis is to create a denoising solution for classification and estimation of BPSK modulated signals with good accuracy and to improve the performance further by tuning the hyper parameters.

We reached a mean squared error of 4.25e-04.

INTRODUCTION

In recent years, machine learning has become an important research area providing us with for example self-driving cars, speech recognition and improved understanding of the human genome. The basic concept of machine learning is to use algorithms to convert these codes into machine language data, learn and from that be able to make predictions about the subject studied. Instead of having to hard-code all procedures to be able to perform a certain task, a large amount of data is fed to the machine to make it capable of learning how to perform the task. Deep learning is an area within machine learning research and a subset of artificial intelligence, Figure 1.

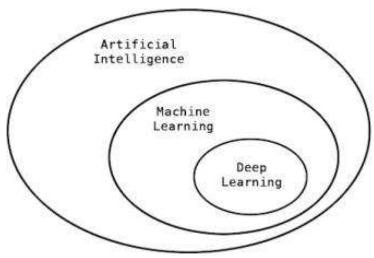


Figure 1: An overview over how artificial intelligence, machine learning and deep learning are related

Deep learning uses a model of computing, called a neural network, inspired by the structure of the human brain. The neural networks consist of neurons, which are the fundamental units of the network. In the neural network the neurons are organized in layers. The neurons in the bottom layer receive the input and the neurons in the top layers are connected to the key. Real-world applications for deep learning algorithms are for example self-driving cars, image classification and signal processing.
In this report, we train a Denoising Autoencoder (DAE) for the classification and estimation of BPSK modulated symbols.

Problem Statement: Classification and Estimation of BPSK Modulated Symbols. The purpose of this report is to propose a denoising solution for a wireless communication system to recreate the original signal from the noisy signal obtained from the system.

Motivation: A Denoising Autoencoder is a modification on the autoencoder to prevent the network learning the identity function. Specifically, if the autoencoder is too big, then it can just learn the data, so the output equals the input, and does not perform any useful representation learning or dimensionality reduction. Denoising autoencoders solve this problem by corrupting the input data on purpose, adding noise or masking some of the input values.

THEORY

Deep Learning

Deep learning is a machine learning technique that teaches computers to do what comes naturally to humans: learn by example. Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. Learning can be supervised, semi-supervised or unsupervised.

Most modern deep learning models are based on artificial neural networks, specifically convolutional neural networks (CNNs), although they can also include propositional formulas or latent variables organized layer-wise in deep generative models such as the nodes in deep belief networks and deep Boltzmann machines. In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input may be a matrix of pixels; the first representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode a nose and eyes; and the fourth layer may recognize that the image contains a face. Importantly, a deep learning process can learn which features to optimally place in which level on its own. This does not eliminate the need for hand-tuning; for example, varying numbers of layers and layer sizes can provide different degrees of abstraction.

The word "deep" in "deep learning" refers to the number of layers through which the data is transformed. More precisely, deep learning systems have a substantial credit assignment path (CAP) depth. The CAP is the chain of transformations from input to output. CAPs describe potentially causal connections between input and output. For a feedforward neural network, the depth of the CAPs is that of the network and is the number of hidden layers plus one (as the output layer is also parameterized).

Autoencoder

An autoencoder is a type of artificial neural network used to learn efficient codings of unlabeled data (unsupervised learning). The encoding is validated and refined by attempting to regenerate the input from the encoding. The autoencoder learns a representation (encoding) for a set of data, typically for dimensionality reduction, by training the network to ignore insignificant data (“noise”).

An autoencoder has two main parts: an encoder that maps the input into the code, and a decoder that maps the code to a reconstruction of the input.

The simplest way to perform the copying task perfectly would be to duplicate the signal. Instead, autoencoders are typically forced to reconstruct the input approximately, preserving only the most relevant aspects of the data in the copy.

Denoising Autoencoders (DAE)
DAEs take a partially corrupted input and are trained to recover the original undistorted input. In practice, the objective of denoising autoencoders is that of cleaning the corrupted input, or denoising. Two assumptions are inherent to this approach:

- Higher level representations are relatively stable and robust to the corruption of the input;
- To perform denoising well, the model needs to extract features that capture useful structure in the input distribution.

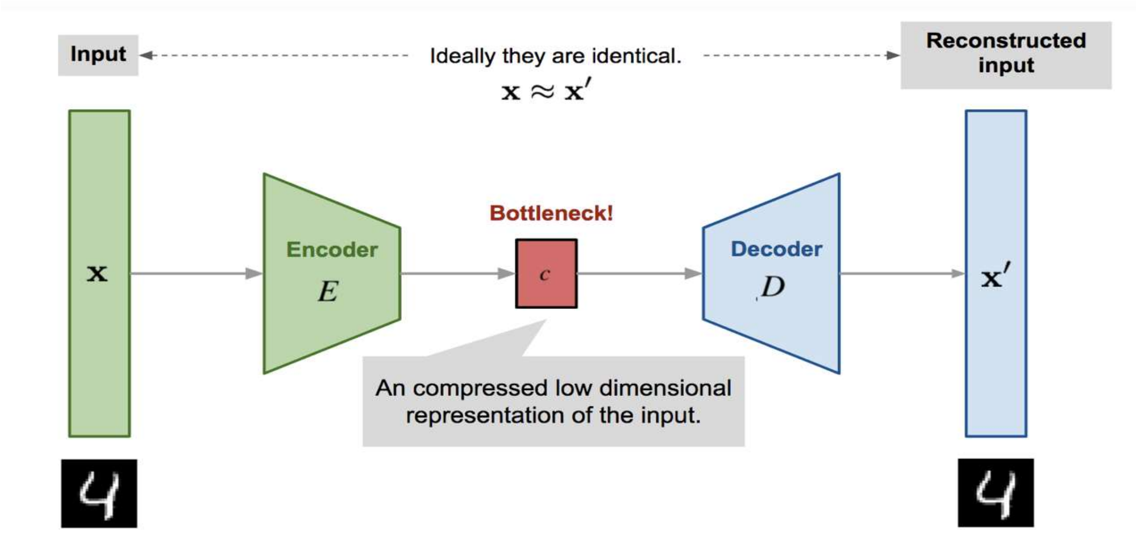
The training process of a DAE works as follows:

- The initial input x is corrupted through $\bar{x}$ through stochastic mapping $\bar{x} \sim q_D\,(\bar{x} \mid x)$.
- The corrupted input $\bar{x}$ is then mapped to a hidden representation with the same process of the standard autoencoder, $h = f_\theta(\bar{x}) = s(W\bar{x} + b)$.
- From the hidden representation the model reconstructs, $z = g_{\theta'}(h)$.

The model's parameters θ & θ' are trained to minimize the average reconstruction error over the training data, specifically, minimizing the difference between z and the original uncorrupted input x. Note that each time a random example x is presented to the model, a new corrupted version is generated stochastically on the basis of $q_D(\bar{x}|x)$.

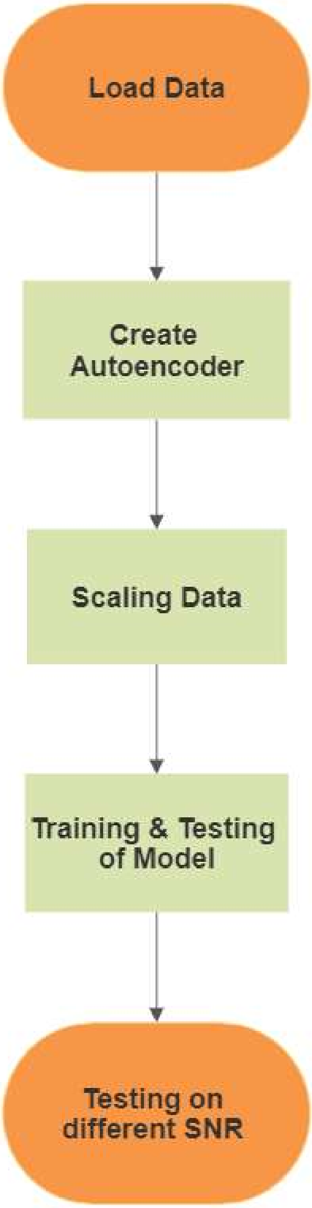
The above-mentioned training process could be applied with any kind of corruption process. Some examples might be additive isotropic Gaussian noise, masking noise (a fraction of the input chosen at random for each example is forced to 0) or salt-and-pepper noise (a fraction of the input chosen at random for each example is set to its minimum or maximum value with uniform probability).

The corruption of the input is performed only during training. After training, no corruption is added.

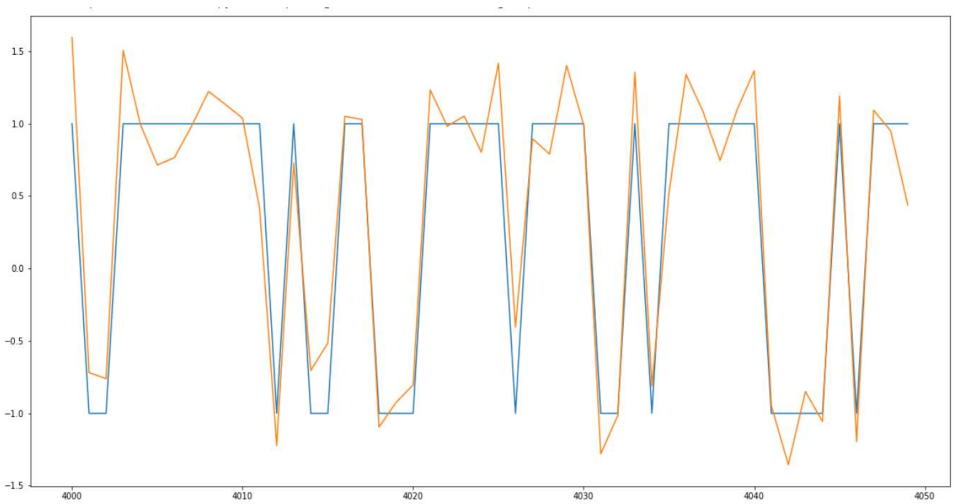


METHODOLOGY & IMPLEMENTATION

We create an autoencoder model that is trained on noisy signals as input and the original signal as the output. The training data is simulated as BPSK modulated symbols at 10dB SNR.



We first load the data which is 100000 BPSK modulated symbols into packets of 16 to be able to train the model as input and output pairs.



Original Data

We then split the data into train, test and validation splits in the ratio 60:30:10 for better results.

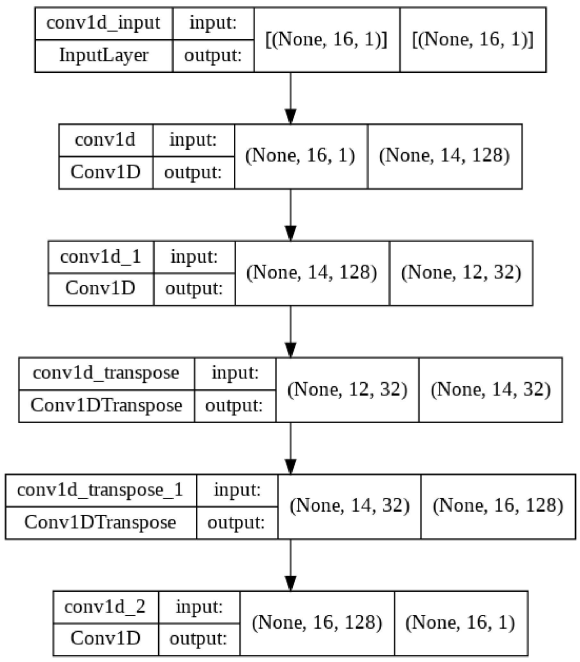
Now we create the autoencoder model with the following parameters:

```
# Model configuration
input_shape = (16, 1)
batch_size = 10
no_epochs = 35
train_test_split = 0.4
validation_split = 0.1
verbosity = 1
max_norm_value = 2.0
```

Here are some insights about the model configuration:

- The input_shape, in line with Conv1D input, is thus (16, 1).
- The batch size is 10. This number seemed to work well, offering a nice balance between loss value and prediction time.

- The number of epochs is fairly low, but pragmatic: the autoencoder did not improve substantially anymore after this number.
- We use 40% of the total data as testing data.
- 10% of the training data will be used for validation purposes.
- All model outputs are displayed on screen, with verbosity mode set to True.
- The max_norm_value is 2.0. This value worked well in a different scenario, and slightly improved the training results.



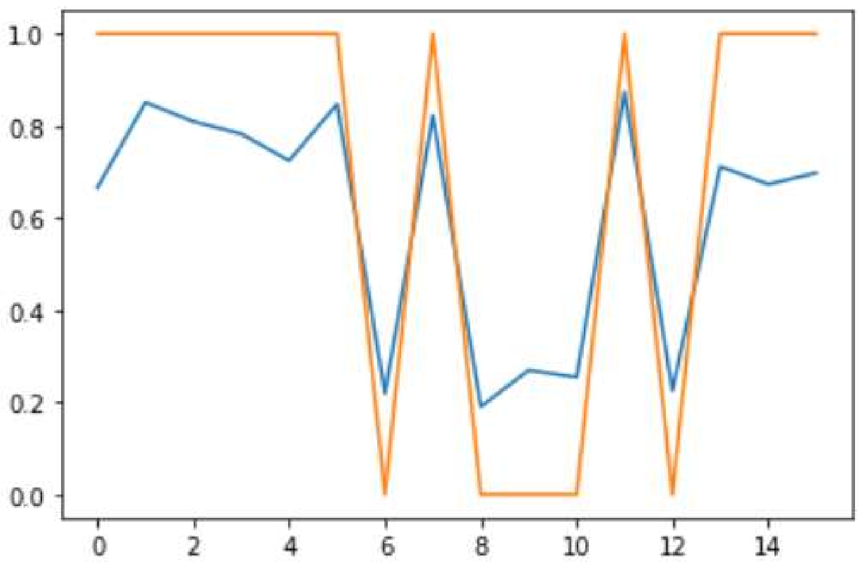
Model Architecture

- We'll use the Sequential API, for stacking the layers on top of each other.
- The two Conv1D layers serve as the encoder, and learn 128 and 32 filters, respectively. They activate with the ReLU activation function, and by consequence require He initialization. Max-norm regularization is applied to each of them.
- The two Conv1DTranspose layers, which learn 32 and 128 filters, serve as the decoder. They also use ReLU activation and He initialization, as well as Max-norm regularization.
- The final Conv layer serves as the output layer, and does (by virtue of padding='same') not alter the shape, except for the number of channels (back into 1).
- Kernel sizes are 3 pixels.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv1d (Conv1D)             (None, 14, 128)           512

 conv1d_1 (Conv1D)           (None, 12, 32)            12320

 conv1d_transpose (Conv1DTra  (None, 14, 32)           3104
 nspose)

 conv1d_transpose_1 (Conv1DT  (None, 16, 128)          12416
 ranspose)

 conv1d_2 (Conv1D)           (None, 16, 1)             385

=================================================================
Total params: 28,737
Trainable params: 28,737
Non-trainable params: 0
_________________________________________________________________
```

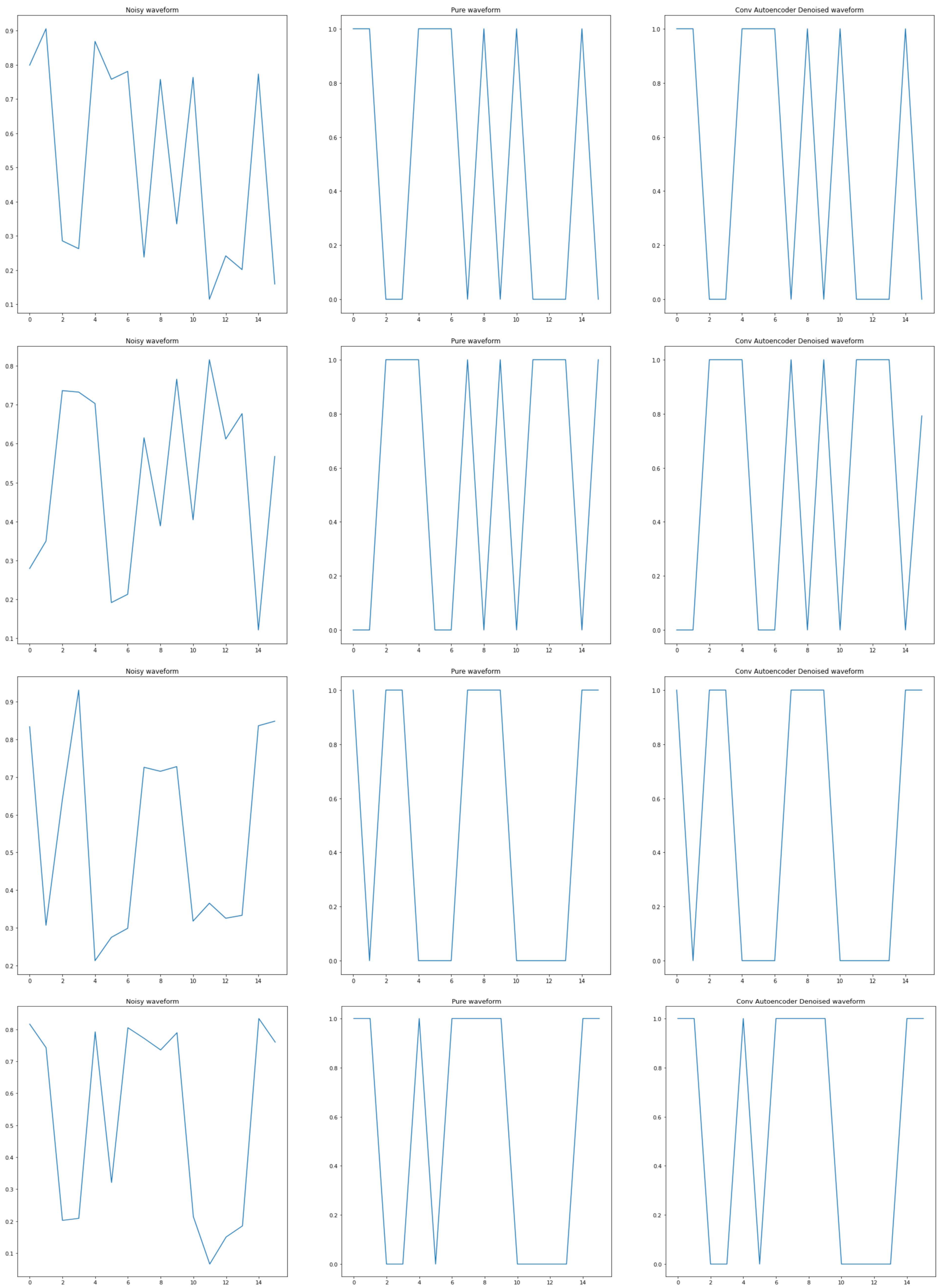
Model Summary

The next thing to do is to compile the model (i.e., specify the optimizer and loss function) and to start the training process. We use Adam and Binary cross entropy for the fact that they are relatively default choices for today's deep learning models. Given the way binary cross entropy loss works, we normalize our samples to fall in the range [0, 1]. Without this normalization step, odd loss values (extremely negative ones, impossible with BCE loss) start popping up.



Scaled Data

Now we train the model on the scaled data. Once the training process finishes, it's time to find out whether our model actually works. We do so by generating a few reconstructions: we add a noisy sample from the test set (which is data the model has never seen before) and visualize whether it outputs the noise-free shape.


Noisy waveform
Pure waveform
Conv Autoencoder Denoised waveform
Noisy waveform
Pure waveform
Conv Autoencoder Denoised waveform
Noisy waveform
Pure waveform
Conv Autoencoder Denoised waveform
Noisy waveform
Pure waveform
Conv Autoencoder Denoised waveform

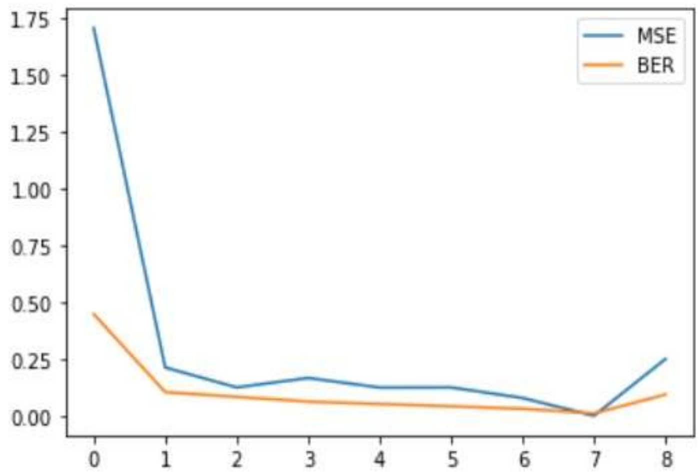
RESULTS

The model was trained for 35 epochs which gave a mean squared error of 4.25e-04 and a mean absolute error of 5.78e-04.

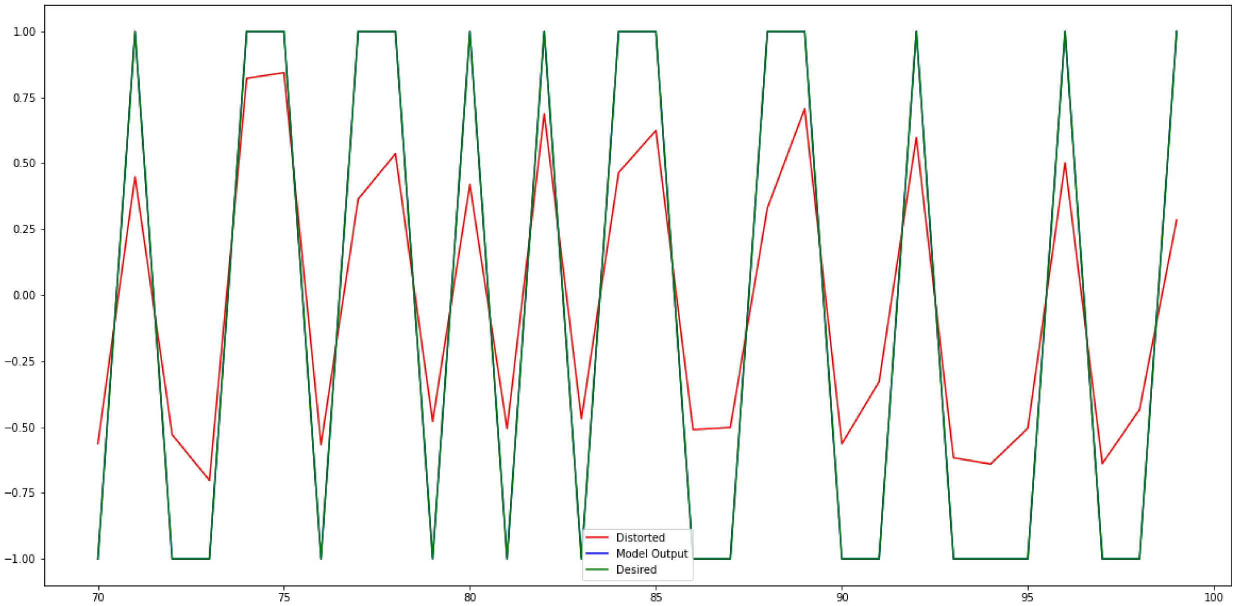
Mean Absoulute Error	Mean Squared Error
0.000578	0.000425

On testing the model over signals of different SNR these were the results:

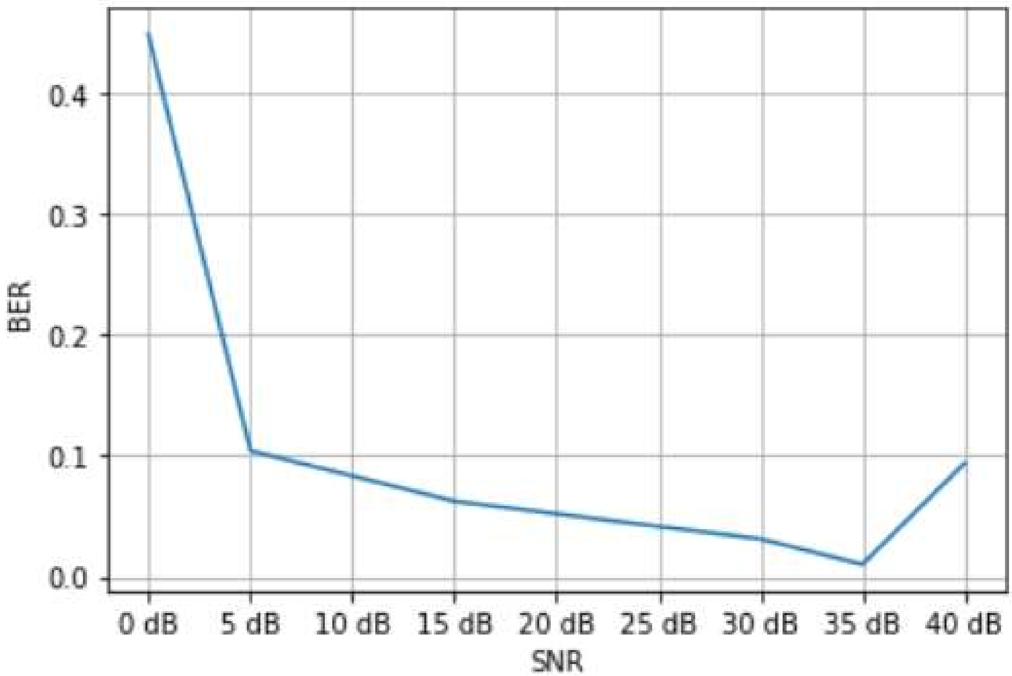
	SNR	MSE	BER
0	0 dB	1.708476e+00	0.447917
1	5 dB	2.128371e-01	0.104167
2	10 dB	1.250001e-01	0.083333
3	15 dB	1.666104e-01	0.062500
4	20 dB	1.246733e-01	0.052083
5	25 dB	1.250000e-01	0.041667
6	30 dB	7.857912e-02	0.031250
7	35 dB	1.813364e-15	0.010417
8	40 dB	2.501442e-01	0.093750



Model Output



BER vs SNR



CONCLUSION

This project considers associate degree implementation of the classification and estimation of BPSK modulated signals task. The proposed model gives near perfect reconstructions of the signal over different SNR. The planned technique was a Deep Learning approach that used an Denoising Autoencoder with keras layers.

REFERENCES

1. A. Sharma, S. Jain, R. Mitra and V. Bhatia, "Performance Analysis of OTFS Over Multipath Channels for Visible Light Communication," 2020 International Conference on Communications, Signal Processing, and their Applications (ICCSPA), 2021, pp. 1-6, doi: 10.1109/ICCSPA49915.2021.9385714.
2. https://github.com/christianversloot/machine-learning-articles/blob/main/creating-a-signal-noise-removal-autoencoder-with-keras.md
3. Noise Learning Based Denoising Autoencoder, https://arxiv.org/abs/2101.07937