Autoencoders-for-dimensionality-reduction

A simple, single hidden layer example of the use of an autoencoder for dimensionality reduction

A challenging task in the modern 'Big Data' era is to reduce the feature space since it is very computationally expensive to perform any kind of analysis or modelling in today's extremely big data sets. There is variety of techniques out there for this purpose: PCA, LDA, Laplacian Eigenmaps, Diffusion Maps, etc...Here I make use of a Neural Network based approach, the Autoencoders. An autoencoder is essentially a Neural Network that replicates the input layer in its output, after coding it (somehow) in-between. In other words, the NN tries to predict its input after passing it through a stack of layers. The actual architecture of the NN is not standard but is user-defined and selected. Usually it seems like a mirrored image (e.g. 1st layer 256 nodes, 2nd layer 64 nodes, 3rd layer again 256 nodes).

In this simple, introductory example I only use one hidden layer since the input space is relatively small initially (92 variables). For larger feature spaces more layers/more nodes would possibly be needed. I am reducing the feature space from these 92 variables to only 16. The AUC score is pretty close to the best NN I have built for this dataset (0.753 vs 0.771) so not much info is sucrificed against our 5-fold reduction in data.

After building the autoencoder model I use it to transform my 92-feature test set into an encoded 16-feature set and I predict its labels. Since I know the actual y labels of this set I then run a scoring to see how it performs.

The data set used is the UCI credit default set which can be found here:https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

AutoEncoder-Based-Communication-System

Implementation and result of AutoEncoder Based Communication System From Research Paper : "An Introduction to Deep Learning for the Physical Layer" http://ieeexplore.ieee.org/document/8054694/

This Repo is effictively implementation of AutoEncoder based Communication System From Research Paper "An Introduction to Deep Learning for the Physical Layer" written by Tim O'Shea and Jakob Hoydis.During My wireless Communication Lab Course,I worked on this research Paper and re-generated result of this research Paper. Idea of Deep learning Based Communication System is new and there is many advantages of Deep learning based Communication.This paper gives complete different apporach than many other paper and tries to introduce deep learning in physical layer.

Abstract of Research Paper

We present and discuss several novel applications of deep learning for the physical layer. By interpreting a communications system as an autoencoder, we develop a fundamental new way to think about communications system design as an end-to-end reconstruction task that seeks to jointly optimize transmitter and receiver components in a single process. We show how this idea can be extended to networks of multiple transmitters and receivers and present the concept of radio transformer networks as a means to incorporate expert domain knowledge in the machine learning model. Lastly, we demonstrate the application of convolutional neural networks on raw IQ samples for modulation classification which achieves competitive accuracy with respect to traditional schemes relying on expert features. This paper is concluded with a discussion of open challenges and areas for future investigation.

From "An Introduction to Deep Learning for the Physical Layer" http://ieeexplore.ieee.org/document/8054694/ written by Tim O'Shea and Jakob Hoydis

Requirements

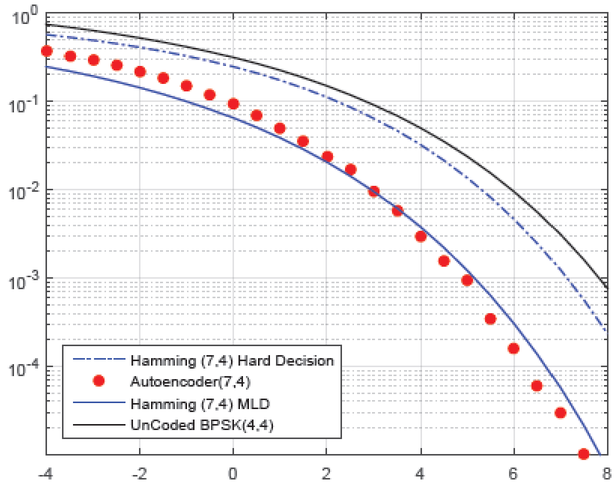
- Tensorflow
- Keras
- Numpy
- Matplotlib

Note

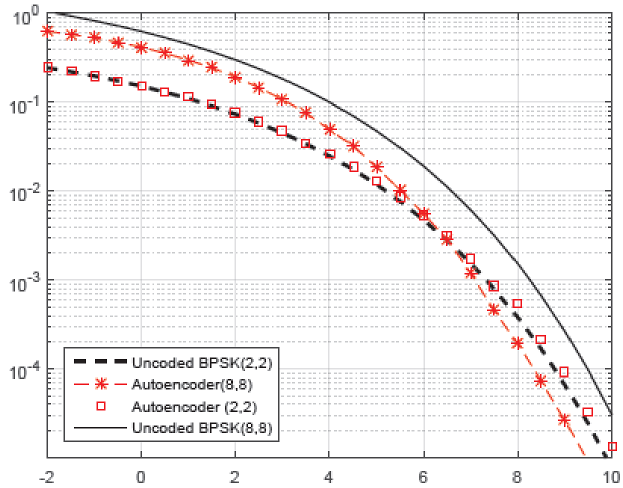
Given Jupyter-Notebook file is dynamic to train any given (n,k) autoencodeer but for getting optimal result one has to manually tweak learning rate and epochs. Plots are generated by matlab script which for now i am not providing it.Anyone can plot result in matlab by training autoencoder and copy-pasting BER array and ploting it into matlab. All re-generated result below are generated with autoencoder_dynamic.ipynb file.

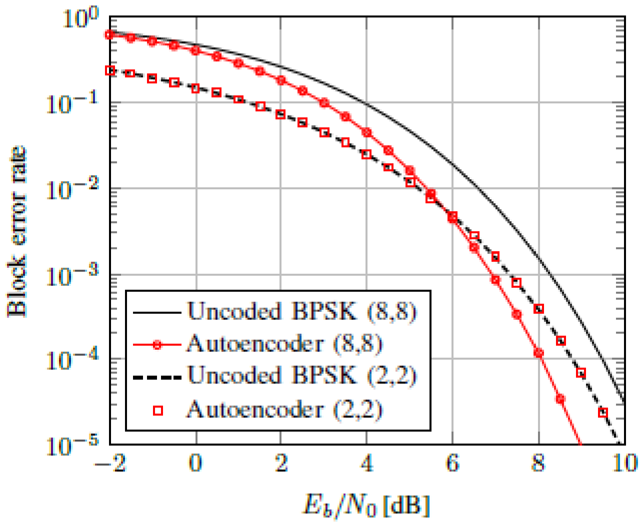
Result

Re-generated Result

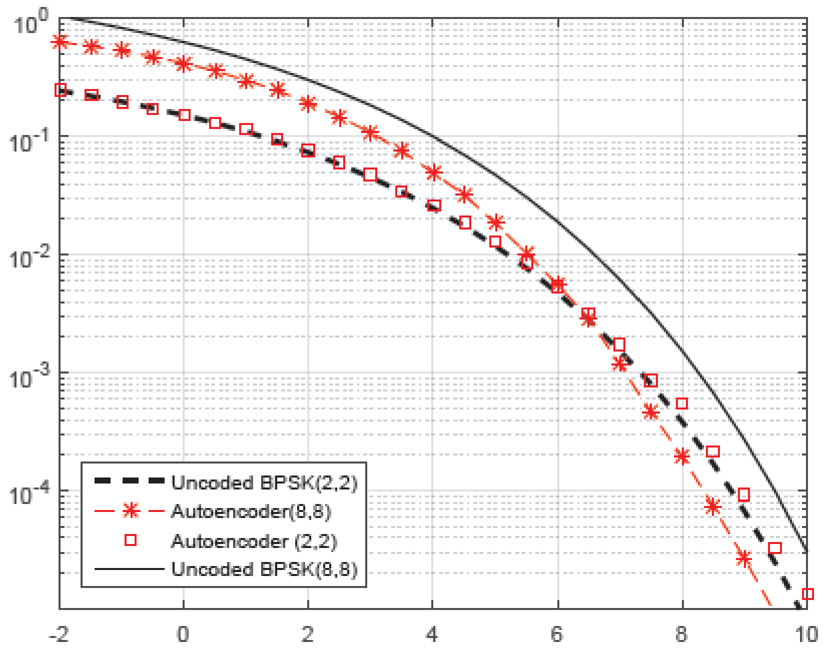


Research Paper





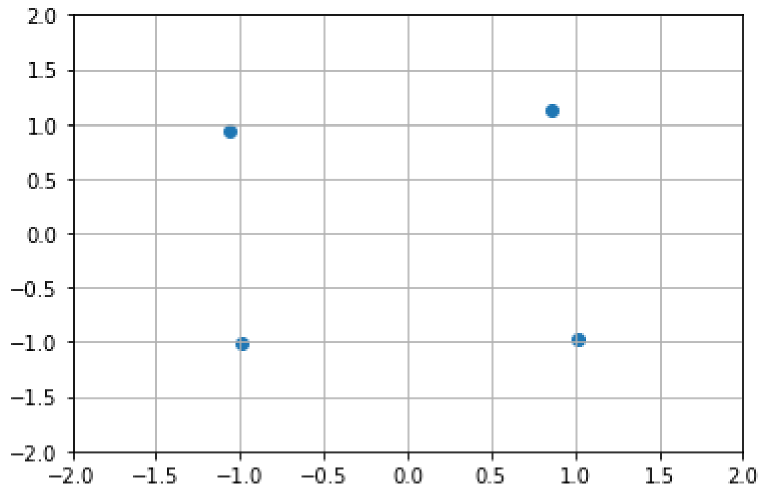
(b)



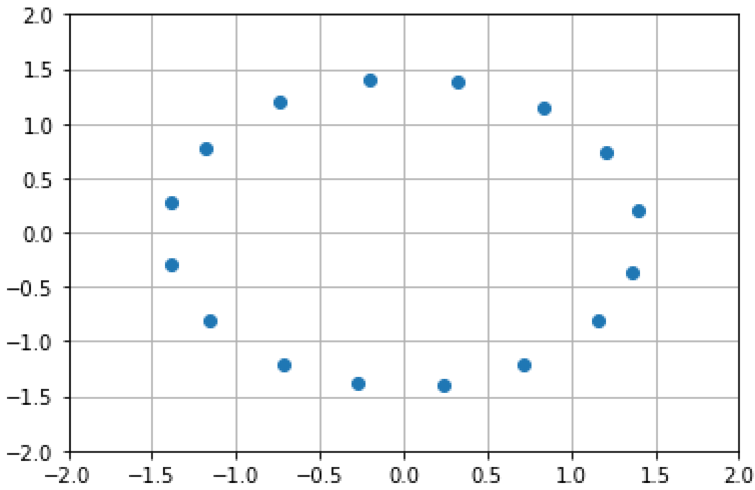
Constellation diagram

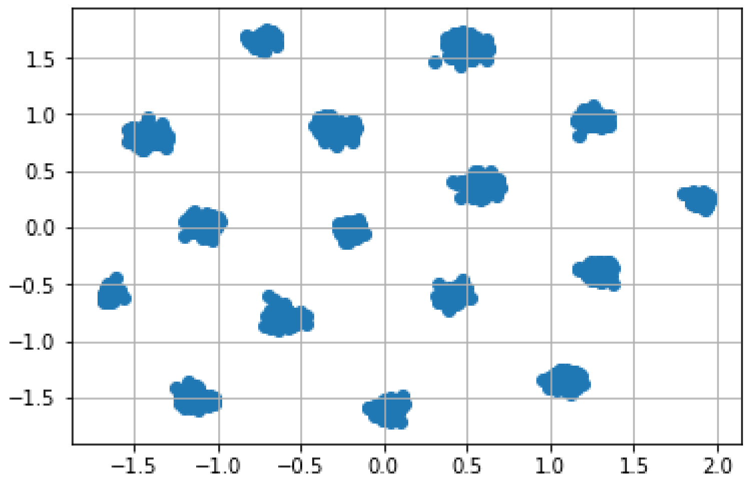
(2,2) AutoEncoder's Constellation diagram

Following Constellation diagram are learned by Autoencoder after training it.



(2,4) AutoEncoder Constellation diagram





About Me

Name : Kiros Gebremariam. This work was done by me during wireless communication Lab Project where I chosed project in research category.

license