Deep Networks for Equalization in Communications

by

Laura Brink

A thesis submitted in partial satisfaction of the
requirements for the degree of
Master's of Science

in

Electrical Engineering and Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Anant Sahai, Chair
Professor John Wawrzynek

Fall 2018

The thesis of Laura Brink, titled Deep Networks for Equalization in Communications, is approved:

Chair ____________________ Date __________

____________________ Date __________

____________________ Date __________

University of California, Berkeley

Deep Networks for Equalization in Communications

Abstract

Deep Networks for Equalization in Communications

by

Laura Brink

Master's of Science in Electrical Engineering and Computer Science

University of California, Berkeley

Professor Anant Sahai, Chair

We apply the techniques from meta-learning to the communications domain. Specifically, we explore how equalization techniques can learn how to handle new environments without training on them.

Contents

List of Figures

Chapter 1

Introduction

Signal processing has long relied on well-defined, structured processes and protocols to function. However, in order to move to more robust and adaptive systems, we will need to overhaul these tightly structured processes. We must design new robust and adaptive communications systems.

From an academic perspective, the rise of machine learning tools and processing has allowed us to tackle problems we have not yet been able to, like in image processing. However, we still do not fully understand the reach or the limitations of this technology. In order to study the limitations of machine learning, we must apply them to spaces that we have studied extensively, like communications, and compare them to the well-known baselines.

Most communications systems have three main processes at the receiver; equalization, demodulation, and error-correction. While we will need to design robust forms of all of these processes, we will focus on equalization for the remainder of this paper.

1.1 Motivation

Meta-learning is the idea that algorithms need to be able to 'learn to learn' in order to generalize to different applications. For example, if a robot is trained to pick up coffee mugs, then we want that robot to be able to quickly learn how to pick up water bottles without having to re-train it. Meta-learning was inspired by humans ability to generalize how to learn [11]. Additionally, researchers in neurology have studied how the brain synapses change over time, suggesting that our algorithms will have to change over time to continue learning [1]. We refer the reader to [12] for a survey of meta-learning technologies and to [6] for recent developments in meta-learning algorithms.

In this paper, we want to understand if our neural network based communication processes can learn to learn. As in, can they handle new environments with new data sequeneces without having to re-train for the new variables. The communications application is arguably an easier application of meta-learning than others have been exploring, like image classification [8]. In order to better understand the reach and limitations of meta-learning, we need to apply it to something simpler, like communications. In this paper, we will explore whether or not we can learn to learn to communicate.

1.2 Background

Inter-symbol Interference and Equalization

Inter-symbol interference occurs when we are transmitting over a channel that has some echos. These echos cause the receiver to hear a garbled signal instead of the original signal from the transmitter. This is called inter-symbol interference because the receiver is hearing a combination of symbols across time.

Let $\vec{x} = [x_0, x_1, \ldots x_n]$ be the set of n complex symbols that the transmitter sends over the channel that connects the transmitter to the receiver. Each channel will have different characteristics. Some channels may have echos, others may have delays, often channels will have both. When a channel has echos, this is called a multipath channel because there are multiple paths to reach the receiver. Each path is called a tap. We can characterize a channel by characterizing the taps.

Let $\vec{a} = [a_0, a_1, \ldots a_\ell]$ be the set of characteristic for a multipath channel that has l taps. When a sequence of symbols like $\vec{x}$ is transmitted over this channel, the channel taps are convolved over the sequence. Additionally, there is noise in the system denoted by v_i.

$$\tilde{x}_m = \sum_{i=0}^{\ell} a_i x_{m-i} + v_i \tag{1.1}$$

The receiver will hear a signal that is corrupted by inter-symbol interfence and noise; $\vec{\tilde{x}} = [\tilde{x}_0, \tilde{x_1}, \ldots \tilde{x}_{n+\ell}]$. Receivers must be able to handle garbled signals in order to transmit data in the real world. The process of removing the inter-symbol interference is called equalization. The goal of equalization is to take in a garbled signal and output a signal with minimal inter-symbol interference.

Figure **??** demonstrates the effects of multi-tap channels on a QPSK modulation constellation. We see that under certain channel conditions, like when the two taps are equal, it is very difficult to distinguish between the four constellations. Engineers have built processes to remove inter-symbol interference. First, let's go into the case when the channel characteristics are known.

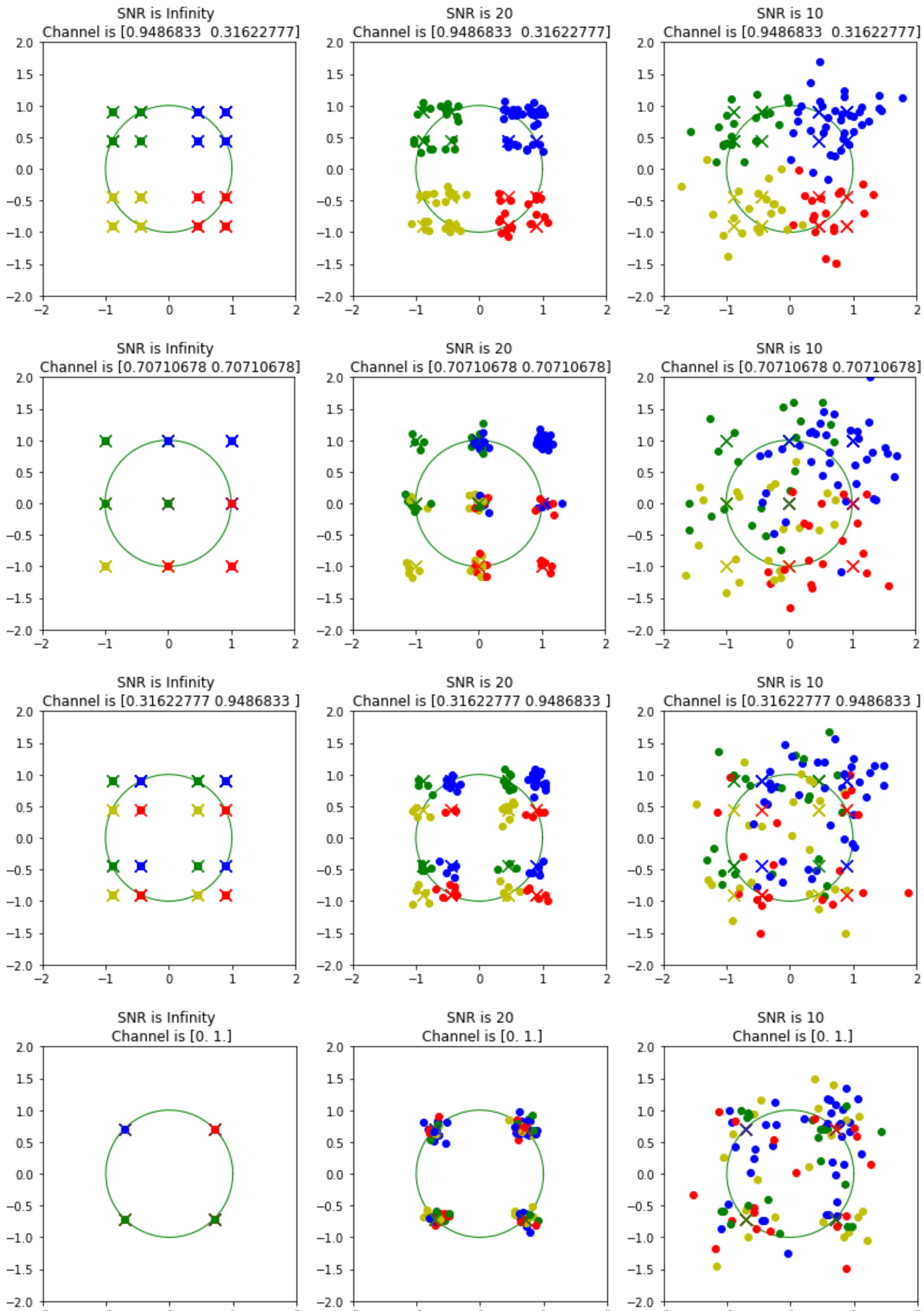
Equalization for a known channel

If you know the channel characteristics, $\vec{a}$, perfectly, then there are a few different methods that can be used.

Zero-forcing

While it's important to consider how well a receiver can equalize with a known channel, this is rarely the case. Usually, we do not know the channel characteristics.

Figure 1.1: The effects of a two tap channel on the QPSK constellation.



Equalization for an unknown channel

When the receiver does not know the channel characteristics, the process of equalization essentially has two jobs; first, identify the channel, second, remove the inter-symbol interference. If the receiver did not identify the channel first, there would be no way to remove the affects of it on the received signal.

In order to do channel estimation, most systems require that packets begin with a known sequence called a preamble. The signal sent will be broken into two parts; $\vec{x} = [\vec{x}_{pre}, \vec{x}_{data}]$. The signal received on the transmitter is $\vec{\tilde{x}} = [\vec{\tilde{x}}_{pre}, \vec{\tilde{x}}_{data}]$. The receiver knows what the orginal preamble sequence was, $\vec{x}_{pre}$, and can use the received preamble sequence, $\vec{\tilde{x}}_{pre}$, to estimate the behavior of the channel. Once the channel is estimated, the receiver then equalizes the data, $\vec{\tilde{x}}_{data}$.

Channel estimation: least squares

$$\min_{H} ||\vec{\tilde{x}}_{pre} - H\vec{x}_{pre}||^2 \tag{1.2}$$

Minimum mean squared error equalizer. We want to minimize the error between the equalized preamble and the original known preamble.

$$\min_{W} ||\vec{x}_{pre} - \vec{\hat{x}}_{pre}||^2 \tag{1.3}$$

$$\vec{\hat{x}}_{pre} = W(H\vec{x}_{pre} + \vec{v}) \tag{1.4}$$

$$W^* = \vec{x}_{pre}\vec{x}_{pre}^T H^T (H\vec{x}_{pre}\vec{x}_{pre}^T H^T + \sigma_v^2 I)^{-1} \tag{1.5}$$

Carrier Frequency Offset and Correction

Now, if we were to implement our minimum mean squared error algorithm on a physical receiver, we would find some problems with our equalization process. Our equalizer will equalize the first symbols very well. However, as we equalize end parts of our sequence, we will encounter a physical phenomonen called carrier frequency offset, CFO.

Carrier frequency offset occurs when ???

When there is a significant CFO present, our symbols will gradually start rotating. CFO will effect our received symbols like

$$\tilde{x}_m = x_m e^{mj\omega} \tag{1.6}$$

The effect will look like something like this

How do real systems handle CFO correction?

There are a few ways to handle CFO, some are more elegant than others.

The first solution is to try to remove the problem. Since CFO is dependent on the length of a packet, one solution is to make packets so short that

Figure 1.2: The effects of a carrier frequency offset on the QPSK constellation.

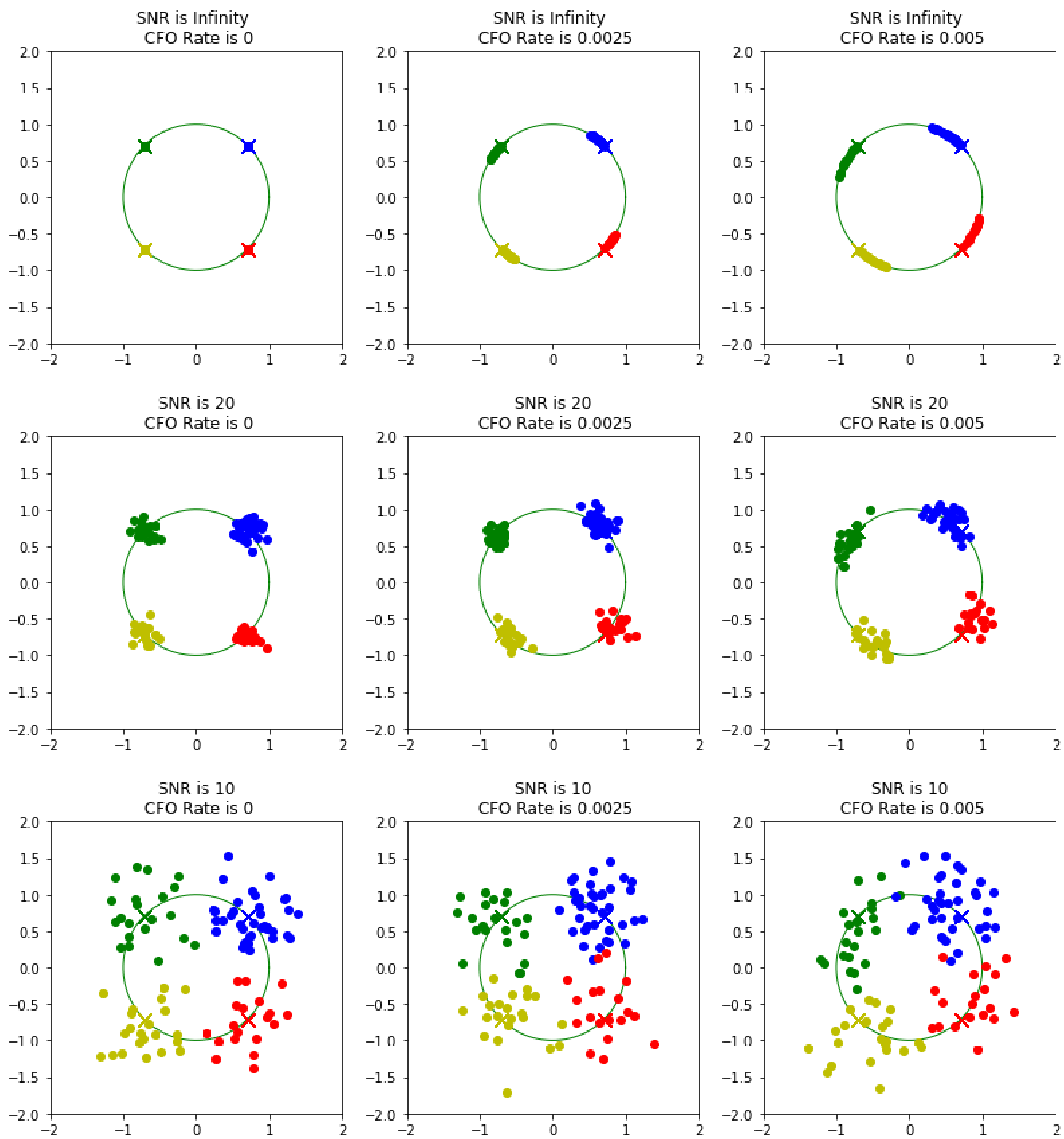
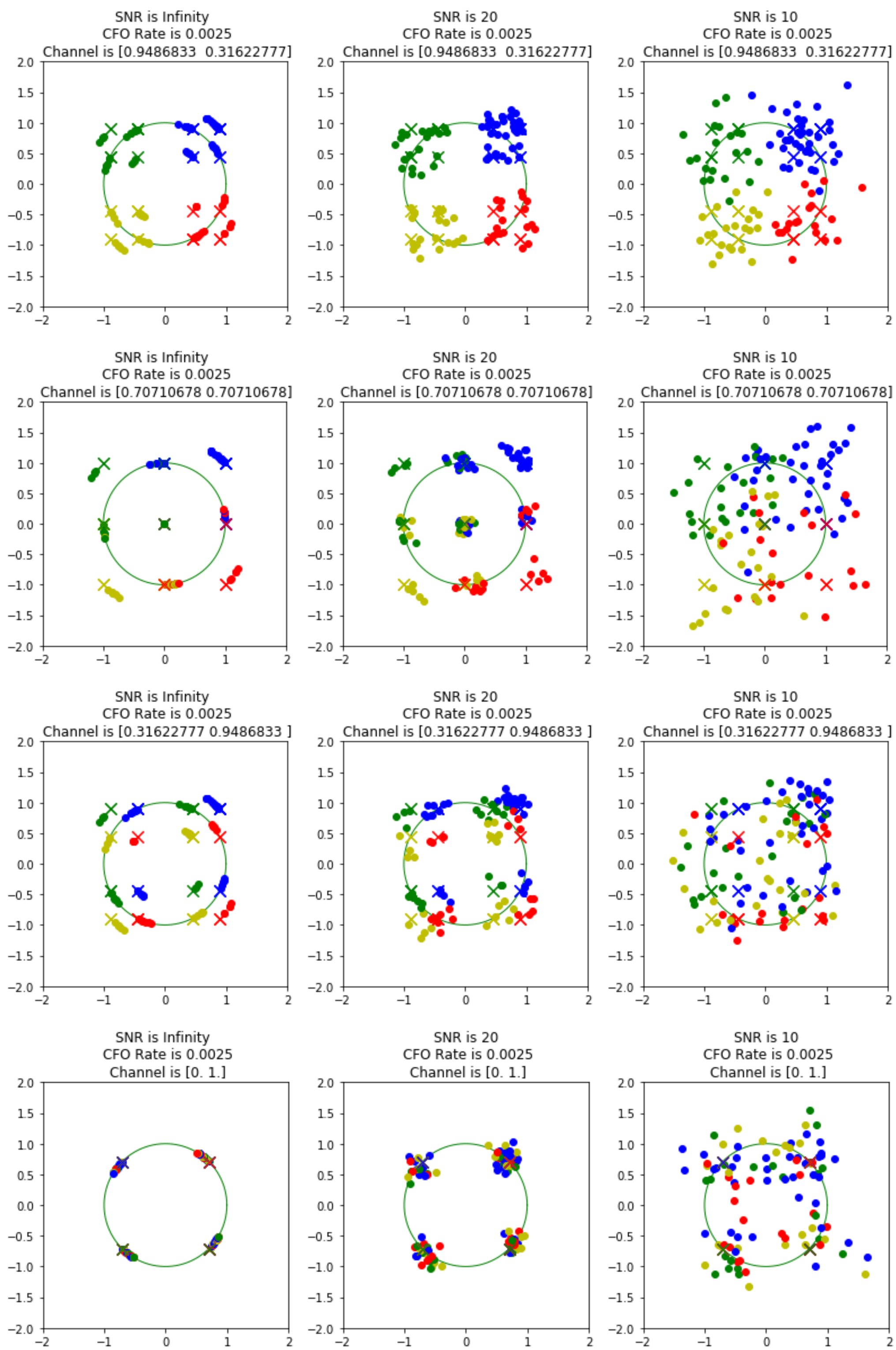


Figure 1.3: The effects of a two tap channel and a carrier frequency offset on the QPSK constellation.



A more elegant solution is using phase-lock loops (costas loops).

What must a modern day receiver handle? What does it look like when we have both CFO and ISI?

$$\tilde{x}_m = (\sum_{i=0}^{l} a_i x_{m-i}) e^{mj\omega} + v_i \tag{1.7}$$

1.3 Related Works

Recently, many researchers have become more interested in applying deep learning techniques to communications systems. We refer the reader to [2][3][20][7][17] for surveys and motivations of these kind of works. Most of the works that we have found typically only deal with additive white gaussian noise (AWGN) channels or Rayleigh fading channels.

In [4], Dörner et. al. implemented an end to end transmitter and reciever with neural networks that allowed gradients to flow all the way back from the receiver during training. However, they restricted themselves to only sending a certain set of messages, only seeing AWGN channels, and they did not address CFO.

Other groups have been focusing on decoding with recurrent neural networks (RNNs) but also only deal with AWGN channels [9][10]. Others have been working with generative adversarial networks (GANs) to train an end-to-end communication system [21]. However, they also only consider AWGN channels or Rayleigh fading channels.

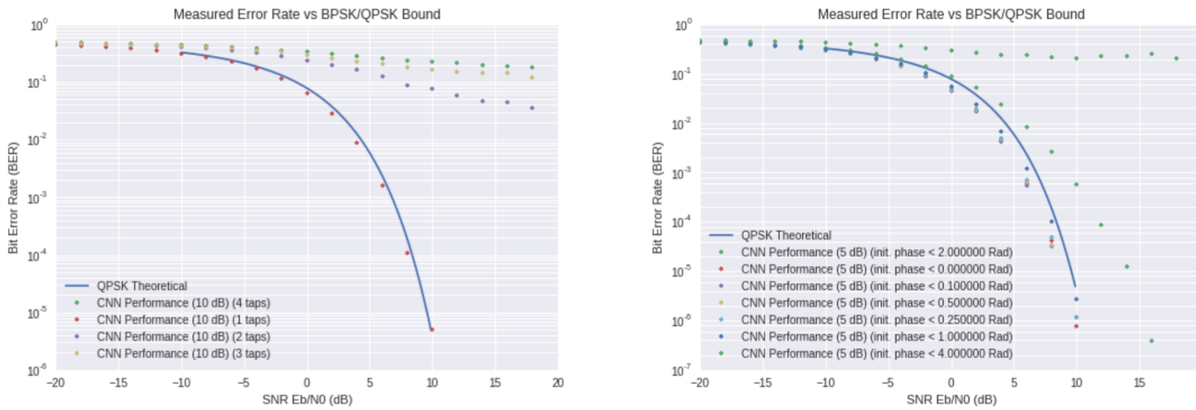
For those that do consider more complex channels, most re-train their models for each new channel seen. Ye et. al. consider OFDM systems where their feedforward neural networks estimate the channel state information then train offline for that specific channel [22]. [19] considers nonlinear channels but re-trains their network for each new channel. Note, this work does not go into detail about the architecture used or how the networks are trained.

Goldsmith and Farsad train a detector for optical and moleculular channels [5]. They assume a Poisson model for the channel and attempt to predict the probability mass function. However, they assume that they retrain for each Poisson parameter and they do not address CFO.

Timothy O'Shea's group has been doing some excellent work in this area. O'Shea's first work in this area jointly optimized a transmitter and receiver for a given channel model (AWGN and Rayleigh fading) but the work does not go into detail about the perforamnce for various channels [17].

In a subsequent paper, O'Shea et. al. explore how multipath channels and random inital phase affects the system performance [13]. Figure **??** shows how their performance drastically decreased with multipath channels and for non-zero phase offset.

In [15], they added an attention model to perform synchronization for time, frequnecy, phase and sample timing offset. However, their work showed that the synchronization was still quite noisy and did not perform much better than without the attention model [13]. Additionally, this work did not consider channel fading.



(a) Impact of Delay Spread

(b) Impact of Random Initial Phase

Figure 1.4: The impacts of multipath channels and initial phase offsets on bit error performance [13].

O'Shea et. al. has also attempted to use GANs to approximate the channel response model for AWGN channels with and without phase noise[14]. The GANs were unable to find accurate probability density functions to represent the channel response.

In [18], they use convolutional neural networks (CNNs) to estimate, but not correct, carrier frequency offset and timing offset. However, they only consider AWGN channels and Rayleigh fading channels. In [16], they explore how unsupervised learning can train autoencoders for multiple antenna communications. They do re-train for each new channel and use a Rayleigh fading channel model.

Chapter 2

Deep Networks for Equalization

2.1 Replicate Results

2.2 Channel Estimation

- compare least squares and how KNN did with pure deep nn based architecture
- NN did better than least squares
- hyperparam search over general 1-layer to 4-layer dense layers, and number of nodes and activations
- plots: how error changed with respect to data points, as number of data points increased, NN outperformed LS
- preamble: 100
- want: QPSK, plot of preamble length
- want? how to visualize that NN does better than LS

Learning an inverse

- can a NN learn to do division?
- given a one tap channel and data sequence, output is equalized data sequence
- with and without log feature scaling
- without log errors = $10^{-}6$
- MMSE gets error = $10^{-}32$

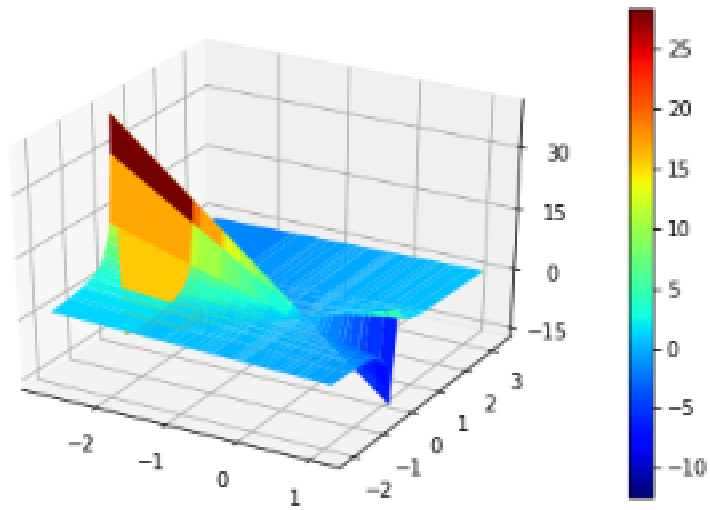


Figure 2.1: Division function; $z = \frac{x}{y}$.

- with log errors =
- straight inversion, without log feature scaling $10^{-}7$ - with dense layers
- inversion with log feature scaling with error of $10^{-}14$
- plots: inversion, but as a function of beta for both non-log and log

Learning to multiply two inputs

NN given x, y - output x*y

2.3 Channel Equalization

- compare to MMSE
- re run with the new RNN architecture
- backprop length of 3. crude search from 1-10. 2 tap channel
- added channel preprocessing: didn't seem to make too much of a difference
- plot log/ log scale to find converging in error

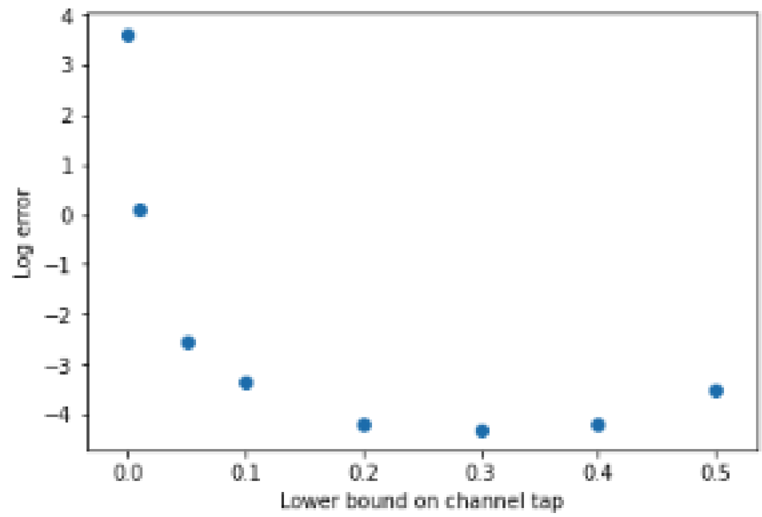


Figure 2.2: One tap channel inversion with respect to range.

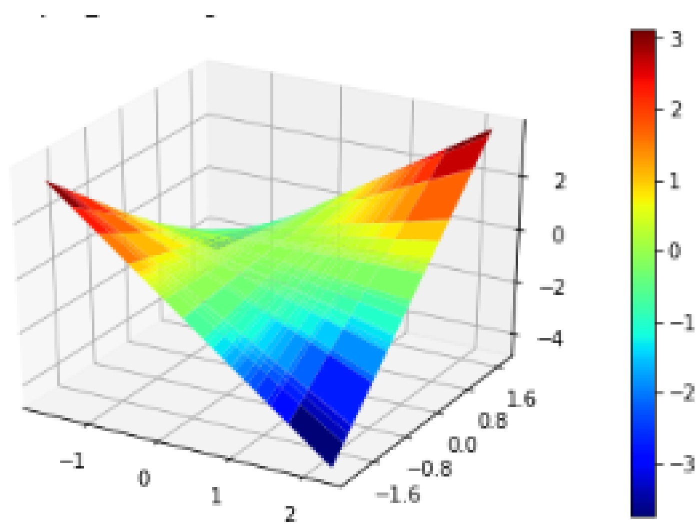


Figure 2.3: Multiplication function; $z = xy$.

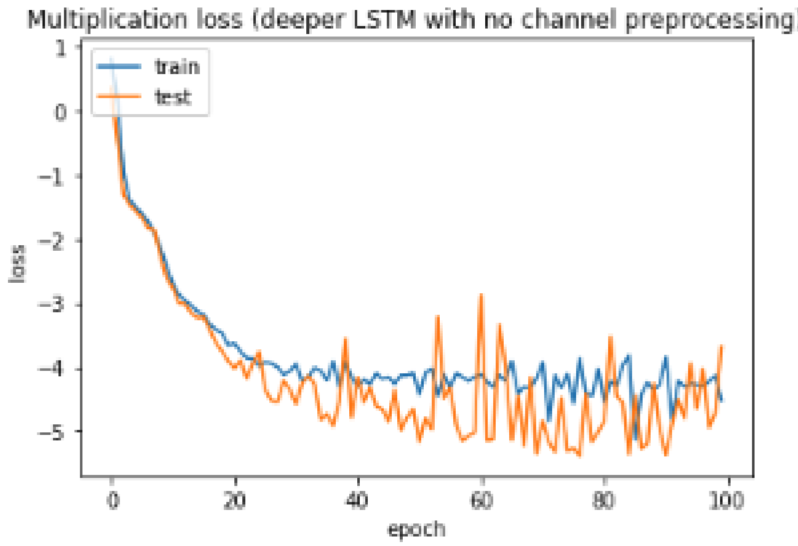


Figure 2.4: LSTM loss trying to learn the Multiplication function; $z = xy$.

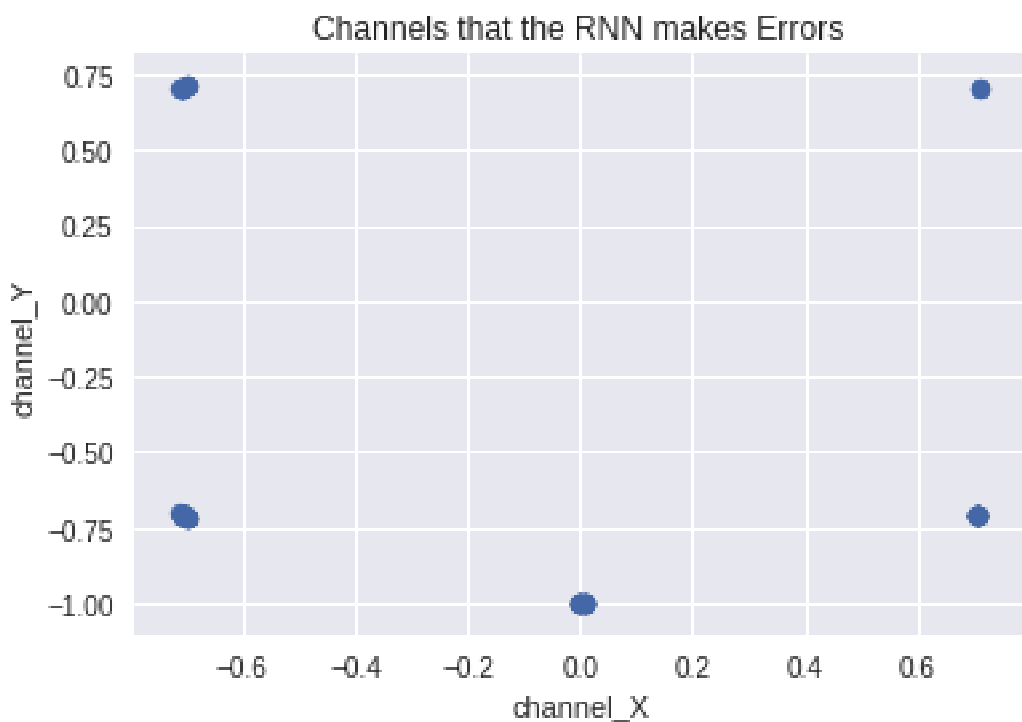


Figure 2.5: What two tap channels does the equalizer get wrong?

2.4 Channel Est + Equal

re run that

Chapter 3

Deep Networks for CFO

3.1 Recurrent Neural Network Follows a Circle

for a constant rate for a given rate

3.2 Deep Network Carrier Frequency Offset Estimation

- complex gradients problems
- act like the real and imaginary parts are separate
- plots: one tap channel plots, without equalization problems
- plots: two tap channel plots, with equalization problems

3.3 Deep Network Carrier Frequency Offset Correction

Program a Costas loop for comparison

Chapter 4

Conclusion

conclucsion

Seria Electronică Şi Telecommuncaţii, Transactions on Electronics and Communications

Bibliography

[1] Yoshua Bengio, Samy Bengio, and Jocelyn Cloutier. "Learning a synaptic learning rule". In: *Technical Report 751, Départment d'Informatique et de Recherche Opérationelle, Université de Montréal, Canada* (1990).

[2] Corina Botoca, Georgeta Budura, and Miclau Nicolae. "Nonlinear Channel Equalization Using Complex Neural Networks". In: *Seria Electronică Şi Telecommuncaţii, Transactions on Electronics and Communications* (2004), pp. 227–231.

[3] Theo Diamandis. "Survey on Deep Learning Techniques for Wireless Communications". In: *Departement of Electrical Engineering, Stanford University* (2017).

[4] Sebastian Dörner et al. "Deep Learning-Based Communication Over the Air". In: *arXiv preprint arxiv:1707.03384* (2017).

[5] Nariman Farsad and Andrea Goldsmith. "Neural Network Detection of Data Sequences in Communication Systems". In: *IEEE Transactions on Signal Processing* PP (2018).

[6] Chelsea Finn, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks". In: *International Conference on Machine Learning* (2017).

[7] Hengtao He et al. "Model-Driven Deep Learning for Physical Layer Communications". In: *arXiv preprint arxiv:1809.06059* (2018).

[8] Siavash Khodadadeh, Ladislau Bölöni, and Mubarak Shah. "Unsupervised Meta-Learning for Few-Shot Image and Video Classification". In: *arXiv preprint arxiv:1811.11819* (2018).

[9] Hyeji Kim et al. "Communication Algorithms via Deep Learning". In: *International Conference on Learning Representations* (2018).

[10] Hyeji Kim et al. "Deepcode: Feedback Codes via Deep Learning". In: 2018.

[11] Brenden M. Lake, Ruslan Salakhutdinov, and Joshua B. Tenenbaum. "Human-level concept learning through probabilistic program induction". In: *Science*. Vol. 350. 6266. 2015, pp. 1332–1338.

[12] Christiane Lemke, Marcin Budka, and Bogdan Gabrys. "Metalearning: a survey of trends and technologies". In: *Artificial Intelligence Review*. Springer, 2015, pp. 117–130.

[13] Timothy J. O'Shea, Kiran Karra, and T. Charles Clancy. "Learning to Communicate: Channel Auto-encoders, Domain Specific Regularizers, and Attention". In: *IEEE International Symposium on Signal Processing and Information Technology*. 2016, pp. 223–228.

[14] Timothy J. O'Shea, Tamoghna Roy, and Nathan West. "Approximating the Void: Learning Stochastic Channel Models from Observation with Variational Generative Adversarial Networks". In: *arXiv preprint arxiv:1805.06350* (2018).

[15] Timothy J. O'Shea et al. "Radio Transformer Networks: Attention Models for Learning to Synchronize in Wireless Systems". In: *arXiv preprint arxiv:1605.00716* (2016).

[16] Timothy O'Shea, Tugba Erpek, and T. Charles Clancy. "Deep Learning-Based MIMO Communications". In: *arXiv preprint arXiv:1707.07980* (2017).

[17] Timothy O'Shea and Jakob Hoydis. "An Introduction to Deep Learning for the Physical Layer". In: *IEEE Transactions on Cognitive Communications and Networking*. Vol. 3. 4. 2017, pp. 563–575.

[18] Timothy O'Shea, Kiran Karra, and T. Charles Clancy. "Learning Approximate Neural Estimators for Wireless Channel State Information". In: *arXiv preprint arXiv:1707.06260* (2017).

[19] K. T. Raghavendra and Amiya K. Tripathy. "An Efficient Channel Equalizer Using Artificial Neural Networks". In: *Neural Network World* 16 (2006), pp. 357–368.

[20] Tianqi Wang et al. "Deep Learning for Wireless Physical Layer: Opportunities and Challenges". In: *China Communications* 14.11 (2017), pp. 92–111.

[21] Hao Ye, Geoffrey Ye Li, and Biing-Hwang Juang. "Channel Agnostic End-to-End Learning based Communication Systems with Conditional GAN". In: *IEEE Global Communications Conference*. 2018.

[22] Hao Ye, Geoffrey Ye Li, and Biing-Hwang Juang. "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems". In: *IEEE Wireless Communications Letters*. 2018, pp. 114–117.