UCLA — Electrical and Computer Engineering Dept.
ECE132A: Communication Systems
Project: Soft Demodulation Using Neural Networks
Due: Friday March 13, 2020

---

1. Please form groups of two people. Make sure that at least one person in the group is reasonably familiar with PYTHON or MATLAB.

2. Generate PYTHON/MATLAB code and all the figures. Please, comment your code! (Remember that in PYTHON anything that follows a # sign is considered a comment. In MATLAB comments are marked by % signs.)

3. Each group should also generate one *short* report. You can use any word processor.

4. The minimum requirements are: report, figures, PYTHON/MATLAB code. If you cannot embed the figures in your document, please zip them with your report. Please indicate clearly the names of the team members.

Project Description

The idea of this computer assignment is to provide some exposure to independent reading, design, and simulation.

1 Introduction

Soft demodulation requires the calculation of log-likelihood ratios (LLRs). When the channel is AWGN, expressions for true or approximated LLRs are well known in the literature. The paper [1] describes the problem and outlines a methodology that can be used to employ neural networks to compute the LLRs.

2 Assignment

1. Read at least up to section III included

2. Consider several modulation schemes in AWGN: 4QAM, 16QAM, 64QAM, BPSK, 8PSK. Write code that computes the exact and approximate LLRs

3. Build a neural network as outlined in [1, §III] and train it. You may want to experiment with different hidden-layer activation functions (e.g., ReLU, SeLU, etc.), different loss functions (e.g., MSE, cross-entropy, etc.), different numbers of neurons in the hidden layer (this number may depend on the modulation order), different training algorithms, different training batch sizes. Do all this for specific SNRs, anywhere between -5 dB and 20 dB.

4. Test your network(s) with new data.

5. What happens if you train the network for a specific SNR but you test it with a different SNR?

6. Try training with data at mixed SNRs and test. How do the results compare to the case where you trained and tested at the same SNR, or with mismatched SNRs?

If you want to do more, here is a suggestions for additional work:

7. Assume that the channel is not AWGN and either you do not want to or do not know how to compute the LLRs. For your simulation, you can assume that in addition to the AWGN, the symbols are multiplied by a Rayleigh-distributed amplitude and phase shifted by a random phase uniformly distributed over $(0, 2\pi)$. For each modulation scheme of order M, build a neural network that tries to classify each symbol (i.e., the network has M outputs, one per symbol). The network outputs reflect the probabilities that each symbol has been sent. Train the network on known symbols and test on a new set of data.

References

[1] O. Shental and J. Hoydis, "'Machine LLRning': Learning to Softly Demodulate," arXiv: 1907.01512v2, 5 July 2019.