A Convolutional Neural Net for Constellation Classification

Alex Knowlton

# Abstract

This work discusses the difficulty of properly demodulating signals in a digital communications system if the transmitted constellation is not known. It is shown that an analytical solution would perform like a searching algorithm and would also require knowledge of the transmitted constellations, which is not a given. A convolutional neural network using the softmax function is shown to have good performance on data without a fading channel. A convolution for further work with a fading channel is proposed.

# Introduction

The field of digital communications uses orthogonal signals for data modulation. Multiple bits are transmitted through expanding the range of phase and amplitude of orthogonal signals so that more than one information bit can be transmitted at a time. However, for effective communication, the transmitted signals must be known before demodulating a received signal. But what if a model could learn what constellation is being transmitted without prior knowledge?

## Digital Communications

To modulate a data signal, information bits are mapped onto M constellation points in a vector space of dimension D (often 2, which is the convention in this report). Each dimension is then mapped onto orthogonal signals before upconversion to a carrier frequency and transmission. With a dimension of 2, the signals can be mapped onto 2 orthogonal signals and then plotted in the IQ plane1 (three different constellations are shown together in figure 1).

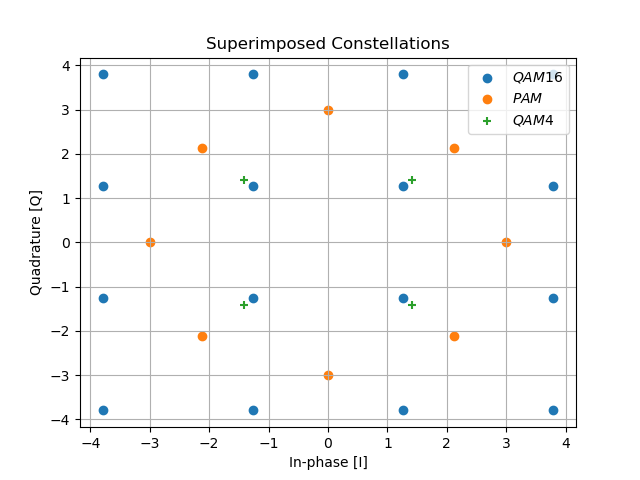


Figure 1: Points from three different constellations superimposed. Note that there is a constant energy per information bit, Eb

The received signals are noisy and often modeled as having passed through a channel, which can be modeled as:

(1)

Where r(t) is the received signal, h is a complex value representing a flat fading channel, c(t) is the transmitted constellation point, and n(t) is zero-mean Gaussian noise1. The received signal, when plotted and compared to the original constellation, might look like figure 2.

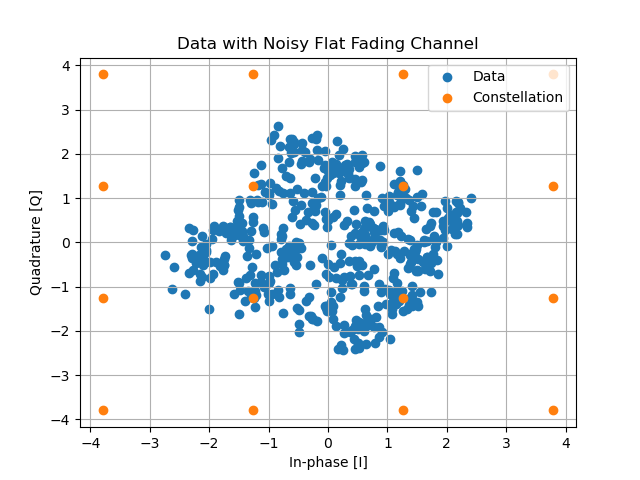


Figure 2: Received signal after being passed through a flat fading channel and corrupted with white Gaussian noise. Note that the received signal is attenuated and rotated.

The reader will note that the more noise there is, the more difficult it is to tell what the original constellation looks like. For example, figure 3 shows two different samples drawn from a QAM 16 constellation with different SNR values.

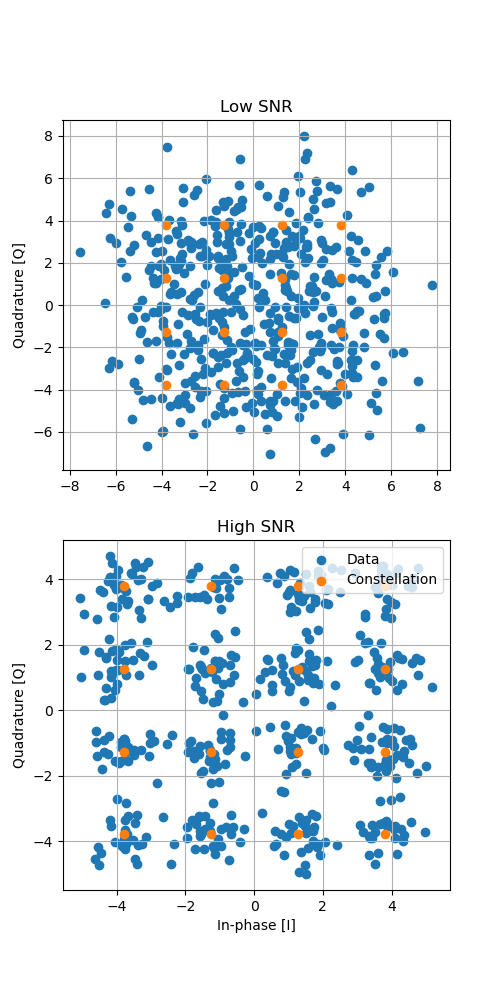


Figure 3: QAM16 data with 2 different SNR values. Note that the low SNR case would be more difficult to classify, since the overall shape is less distinct.

Claude Shannon [2] showed that certain SNR values make it impossible to communicate effectively, so it is clear that certain SNR values would equally make it impossible to clearly classify the received constellation.

## Statistical Hypothesis Testing

Hypothesis testing in statistics is the practice of testing whether sample data supports a hypothesis, usually a hypothesis involving the inference of a population parameter. It can be restated as finding the likelihood that a given sample came from a population.

In this case, the problem in this work can be seen as finding the population most likely to have produced the received signal. The population is a discrete one, but the sample is continuous. The author is not aware of any hypothesis test capable of comparing a continuous distribution to a discrete one.

# Motivation

This work is interesting because, although a convolutional network is useful for the problem, each element in a sample vector is unrelated to the adjacent elements. Therefore, though this problem is, on first inspection, very similar to image classification, the problem is fundamentally different. In images, each pixel is presumed to be related. Sharp lines and differences between adjacent pixels are meaningful and could indicate shapes in the image. In this problem, that is not the case. Therefore, kernels used in convolutions must not span adjacent samples in each signal vector.

As far as the author could tell with a brief survey, this question has not been investigated before in this context, so makes for an interesting challenge, even if the eventual approach turns out to be a relatively simple solution.

# Methodology

A convolutional layer in a neural network calculates almost exactly the required computation to be able to make predictions, but needs to be slightly changed to be effective in this context.

## Analytical Approach

Without a neural network, the approach would be relatively straightforward, though computationally intense. With a given received signal vector and set of constellations, the distance from each point in the signal to each point in each constellation would be calculated, and the signal point would be classified to the point in each constellation to which is has the smallest Euclidean distance. Then, the mean Euclidean distance (or error) associated with each constellation would be calculated, and the constellation with the lowest mean Euclidean distance would be the most likely constellation from which the received signal came (see fig. 4).

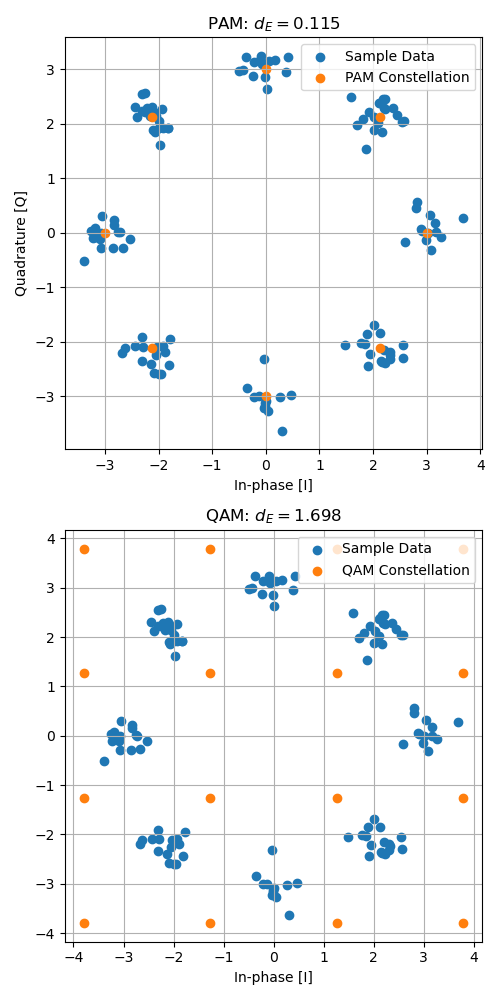


Figure 4: PAM8 signal compared to QAM16 and PAM8 constellations. It is clear that the correct constellation is the one with the lower mean Euclidean distance after classification.

The reader will see that this approach grows very quickly with the size of each constellation, the number of constellations, and the size of the received vector.

## Neural Network Approach

Instead of calculating the distance to each point at once, this work proposes doing it all at once. Returning to figure 1, if each point in the set of all superimposed constellations is correlated with each point in the signal vector, the point with the maximum correlation should indicate the constellation the received vector came from.

The convolution in a convolutional layer can compute the correlation between a point in the received vector and the convolution kernel. If each kernel in a convolutional layer can somehow be assigned an activity score, then the most active kernels will indicate the most likely constellations. Then, simply passing the vector of kernel activity scores through a fully-connected network will give the most likely constellations.

The fundamental trick is that using a softmax layer after the convolution allows a differentiable calculation of activity score. Other works, such as [3], have used kernel activity to explain the kernels in networks, but this work uses it in the classification calculation itself. The use of softmax instead of just max is key, since softmax is differentiable and implemented in PyTorch, so can be easily tested. The proposed network is shown below.

A diagram of a diagram of a rectangular object

Description automatically generated

Figure 5: Proposed neural network architecture. The key is the second layer, which is a sum(softmax) layer, which calculates the activity score for each kernel.

There is one convolutional network with 64 kernels, roughly corresponding to the total number of possible constellation points in all constellations. The convolutional layer is followed by a softmax layer and three fully-connected layers. The network was trained on a simulated dataset with a range of SNR values. The results are shown in the next section.

# Evaluation

The proposed model was trained in PyTorch with a generated dataset, with 30,000 sample vectors drawn as samples from one of 6 constellations with a shape of (512, 2) and then corrupted with a variety of signal-to-noise ratios (SNR). That is, each sample signal has 512 samples, each of which has a real and a complex component.

It is expected that the model would have a harder time with predictions when more noise is present in the data than when less noise is present. Therefore, the model was trained with some noise present in the hope that it would be robust to some received noise.

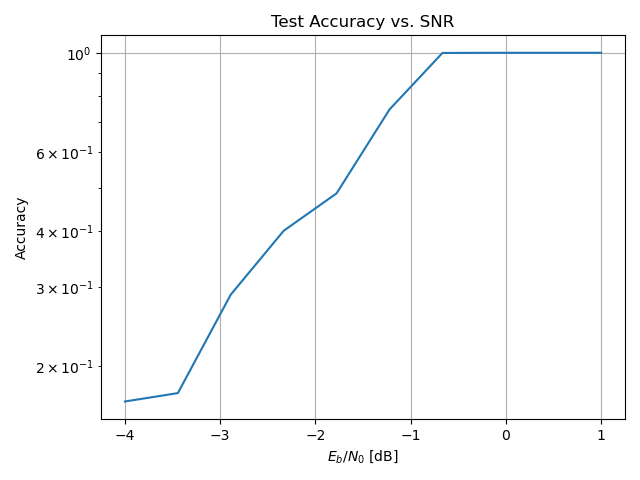


Figure 6: Test accuracy across a range of SNR values. As predicted, noisier data (lower SNR) is harder to classify.

This hypothesis is confirmed; with SNR values above -1 dB, with a standardized energy per information bit (Eb) in each constellation, the model achieves 100% testing accuracy.

# Conclusions

It is shown that a convolutional neural network followed by softmax is effective at constellation classification.

## Challenges

Accounting for a flat fading channel was very difficult. Properly setting a convolution, with padding on each side of the data and setting the stride to 2, can correspond to complex multiplication:

(2)

However, the activation function between a convolutional layer to model the channel and a layer to model the kernel correlation is not clear.

One thing that this work attempted was to model channel fading as well, but since data is only passed through one of 4 channels, some kind of filtering or pooling is necessary.

## Future Work

This project is prime for unsupervised learning. It is possible that a surveillance or emergency response platform could use a similar approach to tap into whatever communication channels are being used on the air and learn how many constellations are present, but also what those constellations are. This work only worked on classification, but it seems feasible that reconstruction of received constellations is possible too.

# References

[1] Stark, Wayne. *Introduction to Digital Communications*. 1st ed., Cambridge University Press, 2023.

[2] Shannon, Claude Elwood, and Warren Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, 1949.

[3] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017, pp. 618-626, doi: 10.1109/ICCV.2017.74.