Machine learning techniques for modulation format recognition

Allyson Silva, Hugo Gomes, Rizan Nejad and Roberto Funes

December 21, 2018

Abstract

In this report, machine learning techniques are used to solve and analyze the modulation format recognition problem. This is a problem in optical communication that consists in defining the type of digital modulation process in which an electrical signal should be sent. A dataset to represent realistic transmission behaviors was generated based on a Gaussian Noise model. A multi-layer perceptron was used and tested with different architectures to show that a high level of accuracy is achievable with machine learning. An analysis of the input features was made by using the *select K best features* method. Finally, an attempt to visualize the data in a 2D was made using the Principal Component Analysis and t-distributed Stochastic Neighbor Embedding methods to reduce the dimensionality of the input features and see their relationships. The code is available here.

1 Introduction

A significant increase in the number of services supported by a network is expected in the following years judging by the observed annual growth in internet traffic. Networks should be capable of adapting to the new conditions that may arise using the knowledge achieved using previous measurements.

Modulation in optical communication is a process of converting an electrical signal into a bit stream with the objective of compressing data as much as possible so that less bandwidth is occupied. Thereafter, it is important to define the modulation format in which the signal is sent given the limited bandwidth available. The modulation format is the type of modulation that defines how fast we are able to send the signal without distorting it. The problem to define the best modulation format is known as modulation format recognition (MFR) (Bo et al., 2016; Ren et al., 2017) and attempts to solve it using machine learning (ML) techniques can also be found in recent literature (Guesmi and Menif, 2015; Wang et al., 2017; Rafique, 2018).

There are different types of modulation. In this project, we focus on digital modulations, which can exchange information at high rates. The modulations considered are Binary Phase Shift Keying (DP-BPSK), Quadrature Phase Shift Keying (DP-QPSK), 16-Quadrature Amplitude Modulation (DP-16QAM), and 64-Quadrature Amplitude Modulation (DP-64QAM). The reverse of modulation used in the transmitter is done at the receiver side.

Since modern high-speed networks have the ability to adapt the modulation format of the transmitted data according to new changes in the communication link, it is highly important to correctly detect it at receiver side. When the modulation format of the transmitting signal is known, it can be demodulated at the receiver with no need to use control commands from control plane or human interference. Otherwise, this would significantly slow down the process. This is undesirable in optical communications since it aims to transmit the data as quickly as possible and with high spectral efficiency. In addition, knowing about the modulation format in advance will be helpful in other techniques such as equalization, which aims to mitigate impairments of the signal transmitted through a channel.

The remainder of this report describes how we used machine learning techniques to deal with the MFR problem. Section 2 describes the dataset generated to simulate a realistic transmission behaviour. Section 3 presents the methodology used in this work, i.e. the ML techniques used and their results. Finally, the conclusion of this work is presented in section 4.

2 Dataset generation

Access to real-world data is hard. First reason is that only large companies have access to them and they do not provide since the data is usually private and their exclusivity is part of their business success. Second, it is difficult to simulate real conditions in lab using current equipments. However, we can model an optical transmission channel, considering its nonlinearities, by using a Gaussian-Noise (GN) model as described in Poggiolini and Jiang (2017). The features and targets drawn from the mathematical expressions are used to simulate a dataset. The model is an approximation of a nonlinear propagation in optical systems. Thus, consequently, the dataset will also be.

For the project, we use 11 features as in Rafique (2018). Their labels and ranges are presented in Table 1. Two input features (*Launch power* and *Channel grid*) are dependent on *Symbol rate*. The formulas to calculate them are also found in Poggiolini and Jiang (2017) and were used in the dataset generation.

Given the input features, the MFR problem objective is to decide in which modulation format classification the signal should be sent. Four possible classifications are considered: DP-BPSK, DP-QPSK, DP-16QAM, and DP-64QAM. The optimal choice is defined taking into account optical and electrical component limitations. Briefly, a back-to-back optical signal-to-noise ratio (OSNR) is used to deter-

Feature	*Range*
Symbol rate	30 - 90
Roll-off	0.01 - 0.05
Launch power	f(symbol rate)
Channel load	1 - 120
Dispersion	4 - 21
Nonlinear index	0.8 - 1.6
Loss	0.15 - 0.2
Span count	1 - 50
Span lenght	40 - 120
Noise figure	4 - 6.5
Channel grid	f(symbol rate)

Table 1: Quantitative ranges for input features generated

mine the best modulation classification option to choose. The OSNR can be approximated applying the GN approach (Poggiolini and Jiang, 2017) for a set of known input features. After classifying each sample into different modulation format values, using the digital communication approach explained in Dias et al. (2017), which takes into account OSNR, back-to-back OSNR and penalty for each format, we managed to obtain its related target. The described optical link setup is shown in Figure 1, which is an adaptation from Rafique (2018).

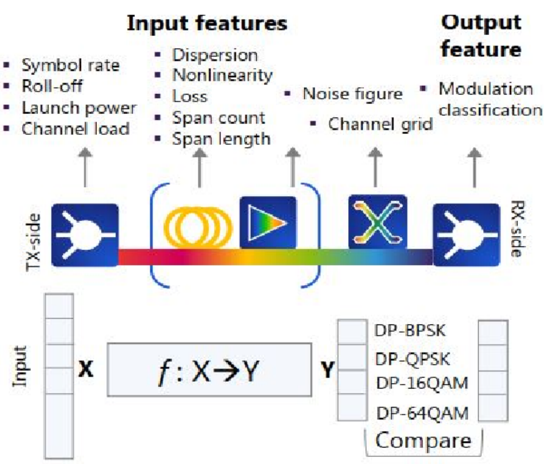


Figure 1: Optical link setup adapted from Rafique (2018)

When generating the dataset, it is possible to choose a combination of input features that will result in an unclassified data to any of the modulations. Since Rafique (2018) did not consider these data in their model, we are rejecting them too. According to them, this synthetic data generation model represents realistic transmission behavior that were attested by several bodies of work. As in Rafique (2018), we generated 100,000 unique combinations of input features that its possible to associate to a classification.

3 Methodology and experiments

The objective of the project is to apply some concepts learned in class to deal with the problem described using the dataset generated. We used three procedures to help us to better understand the problem. The first is a similar replication of the multi-layer perceptron (MLP), as used in Rafique (2018), to solve the problem using the given inputs and outputs. The objective was to compare our methods and verify the consistency of the dataset generated. The second part we used a feature selection algorithm to filter the most relevant features for the classification with the objective to determine those that are more important to companies invest more on measurements, which could help them to apply efficiently their resources for data acquisition. Finally, in the last part we use two dimensionality reduction methods to provide a data visualization of the dataset in a 2D. We think that this would help to understand the reason that the MLP as in Rafique (2018) consistently gets a very high score.

The methods were implemented in Python, using Keras, a high-level neural networks API, written in Python and capable of running on top of TensorFlow, and scikit-learn (machine learning in Python) libraries. The experiments were run on Intel(R) Core(TM) i7-6700K processor.

3.1 Multi-layer perceptron

MLP is used as a feedforward model with supervised learning trained via a backpropagation algorithm using all 11 input features and 1 output feature aforementioned. This output is a multi-categorical variable representing the four modulations. The dataset was split 60% as training data, 20% as validation data and 20% as test data. The performance of the MLP is measured by its score, i.e. the percentage of correct predictions. In all experiments in this work, the input data was standardized to have a mean of 0 and standard deviation of 1.

Four configurations for the MLP were considered: 1:(5), 1:(10), 1:(100) and 2:(100x10). The first number indicates the number of layers while the number between parenthesis indicates the number of neurons in each layer. All layers used rectified linear (ReLu) activation functions, except the output layer which used soft-max activation function. We used a batch size of 100 samples and a learning rate of 0.01, as in Rafique (2018).

The initialization of the MLP is not specified in Rafique (2018), so we had to guess our own way to do. We used *RandomUniform* function from available initializers in Keras to initialize the weights. It generates weights with a uniform distribution between $minval = -0.05$ and $maxval = 0.05$.

Our first set of experiments evaluated the performance of our classifiers with the four configurations tested. The curves in Figure 2 represent the evolution of classification accuracy on validation data according to the number of epochs run (500 epochs). As in Rafique (2018), it can be seen that all architectures follow a similar saturation behavior after a certain number of epochs. We used the early stopping to say that the training converged with the criteria of 5 epochs without improvement on the accuracy in the validation set. Scores on the graph are shown for where convergence was detected minus the five epochs without improvement. The graph shows that convergence occurs earlier at 2:(100x10) than at the others and at a higher score, except for 1:(100) where the same score is achieved but after a higher number of epochs.

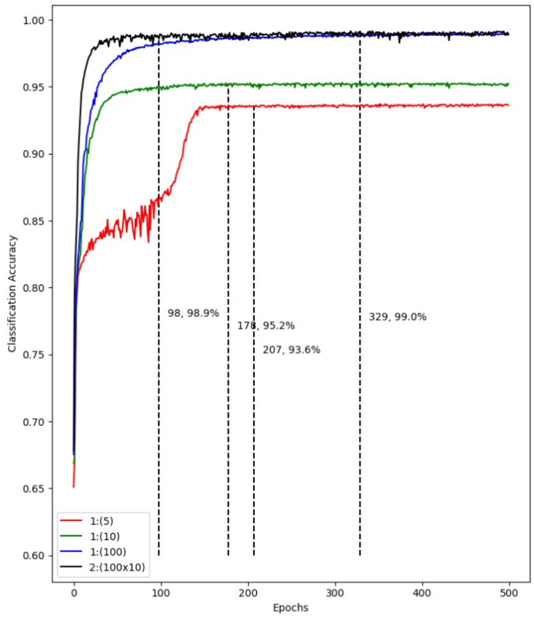


Figure 2: Classification accuracy and number of batches to converge on each MLP setting

To select the best setting, we should also consider the training time of each method. Figure 3 presents the model training time to reach convergence for the four MLP architectures considered. We observe that the time do not change much between each configuration tested, ranging from 150s to 200s, on the computational environment tested. Since the best scores obtained when converging were observed for 1:(100) and 2:(100x10), any of them could be concluded as being the best option.

The confusion matrices in Tables 2-5 provide detailed results on the test data for each of the MLP settings. We observe that most of the errors occurs between classifications DP-BPSK and DP-QPSK. Rafique (2018) explain that these errors may be attributed to very similar cardinality properties for DP-BPSK and DP-QPSK.

The results found were consistently better than those presented in Rafique (2018). The scores ranged from 92.69% to 98.39% between each architecture evaluated in there, while our scores were from 93.82% to 98.68%. Although the results were slightly different, this can be explained on the differences on the dataset used, since we had to generate our own based on their description and some assumptions we had to made about their method that were not on the paper, such as the standardization of the input data. However, the same conclusions they pointed could also be observed in our experiments, showing that our work approximated very well their MLP.

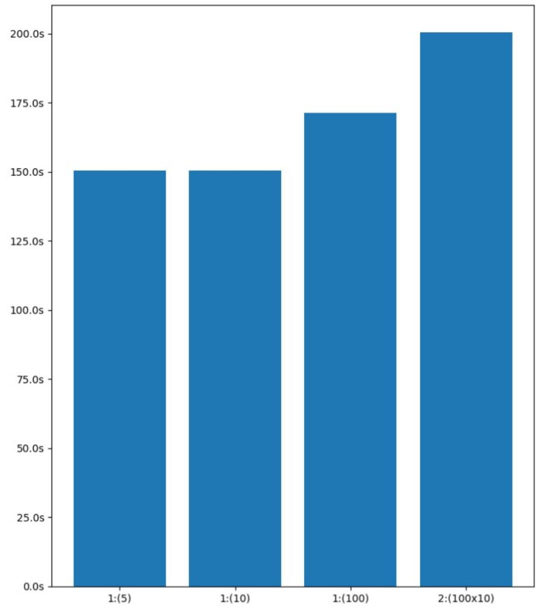


Figure 3: Time to convergence on each MLP setting

1:(5) neurons, avg accuracy = 93.82%				
MLP \ true	DP-BPSK	DP-QPSK	DP-16QAM	DP-64QAM
DP-BPSK	2834	674	0	0
DP-QPSK	582	6674	4	0
DP-16QAM	0	3	6511	0
DP-64QAM	0	0	0	2745
Accuracy	82.96%	91.99%	99.94%	100%

Table 2: Confusion matrix for MLP architecture with 1 hidden layer with 5 neurons

3.2 Feature selection

For the next part of our work, we present a feature analysis done with the help of a feature selection method used in class. The objective is to know if we can reduce the number of features used to choose the modulation classification without losing accuracy on the method. Some features measurement can be quite expensive for a company, while others cannot be measured with certitude. Finding a reduced number of input features while maintaining a good classification score would certainly mean a good achievement in this work.

The *select K best features* method was used. It is a forward sequential selection method which selects K features that achieved the highest scores according to an individual performance measurement. For simplicity, we used the χ^2 test as performance measurement for the features. For each feature added to the set of selected features, we fit the MLP with 2 hidden layers and 100 and 10 neurons to observe at which accuracy level the validation data would converge. Table 6 sort the features according to their χ^2 and show the

1:(10) neurons, avg accuracy = 94.99%				
MLP \ true	DP-BPSK	DP-QPSK	DP-16QAM	DP-64QAM
DP-BPSK	2971	553	0	0
DP-QPSK	445	6771	4	0
DP-16QAM	0	0	6511	0
DP-64QAM	0	0	0	2745
Accuracy	86.97%	92.45%	99.94%	100%

Table 3: Confusion matrix for MLP architecture with 1 hidden layer with 10 neurons

1:(100) neurons, avg accuracy = 98.52%				
MLP \ true	DP-BPSK	DP-QPSK	DP-16QAM	DP-64QAM
DP-BPSK	3165	45	0	0
DP-QPSK	251	7279	0	0
DP-16QAM	0	0	6515	0
DP-64QAM	0	0	0	2745
Accuracy	92.65%	99.39%	100%	100%

Table 4: Confusion matrix for MLP architecture with 1 hidden layer with 100 neurons

2:(100x10) neurons, avg accuracy = 98.68%				
MLP \ true	DP-BPSK	DP-QPSK	DP-16QAM	DP-64QAM
DP-BPSK	3328	174	0	0
DP-QPSK	88	7150	1	0
DP-16QAM	0	0	6514	0
DP-64QAM	0	0	0	2745
Accuracy	97.42%	97.62%	99.98%	100%

Table 5: Confusion matrix for MLP architecture with 2 hidden layer with 100 and 10 neurons

K	*Feature*	χ^2	*MLP accuracy*	*Epochs*
1	Span count	4625.19	61.76%	49
2	Symbol rate	26.10	65.30%	54
3	Channel grid	24.56	65.57%	59
4	Span length	17.09	78.32%	81
5	Launch power	13.79	81.55%	114
6	Noise figure	5.78	86.86%	134
7	Dispersion	4.86	87.56%	127
8	Nonlinear index	2.74	92.20%	100
9	Loss	1.58	94.82%	106
10	Roll-off	0.55	95.72%	62
11	Channel load	0.23	98.68%	98

Table 6: Confusion matrix for MLP architecture with 2 hidden layer with 100 and 10 neurons

accuracy obtained when we run the MLP with up to the K best features and the number of epochs where convergence is reached.

From the table, we observe that all features are relevant if a maximum level of accuracy is desirable. However, depending on the costs to measure some of the features, specially those with the lowest χ^2, it is possible to remove them from the input features and still get a very good accuracy saving resources that could be spent at measurement. In fact, a score near 80% is achievable using as low as 4 features.

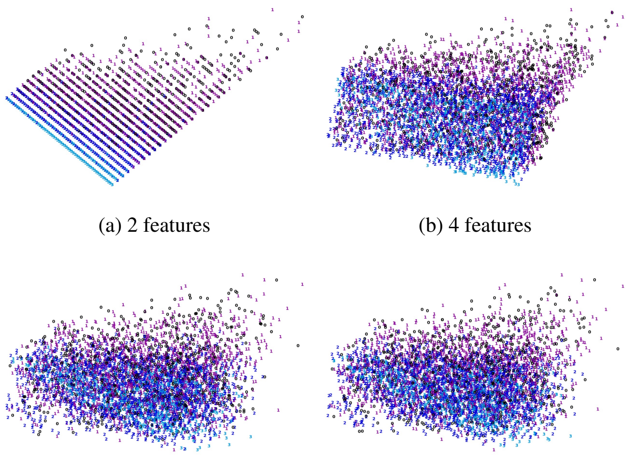
3.3 Data visualization

In order to support the good performance that we achieve and the results of selecting relevant features to solve the problem, we present a visualization of our data in a two-dimensional space. We selected two algorithms to achieve this: Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE). They are used in this work to tackle the problem of curse of dimensionality in our data. Even for a low number of features, it is important to see the relationship between them to try to remove noise and difficulties for the classifier. This part of the work aims to have a better understanding of our data and to support the idea to make it less complex.

PCA is a technique of dimension reduction that can be used to visualize the data in a two-dimensional space by selecting the number of Principal Components as two. PCA only captures linear relationships and two principal components might not be enough to get the percentage of variation that we want to be captured in the dataset. In order to improve the vizualization with PCA, we considered different number of features. Figure 4 presents the data visualization using PCA to reduce the dimensionality of the dataset considering the best 2, 4, 8 and all 11 input features, using the same selection method as presented before.

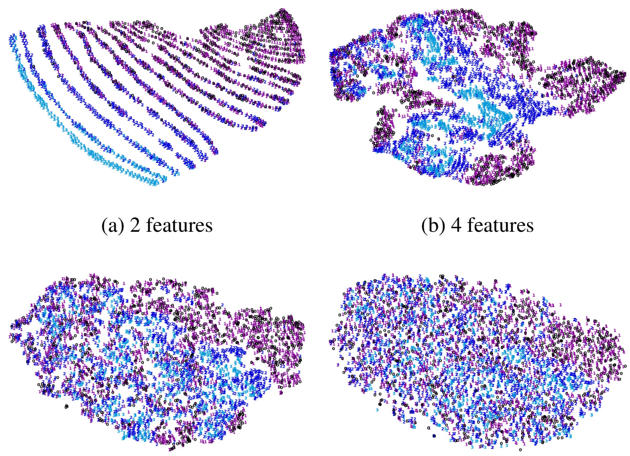
From the figure, we could not find a nice structure behind the data. It might be because PCA only captures linear relationships and our dataset can have non-linear associations. An alternative to capture non-linear relationships is to use t-SNE. t-SNE is also a technique for dimensionality reduction specifically well suited for visualization. Although it does not preserve distances or densities, t-SNE aims to preserve nearest-neighbors and can lead to good visualizations. Figure 5 show the visualization using t-SNE for the same number of input features as used for PCA.

We observe that t-SNE represents better our data in the two-dimensional space, giving good insights about the classification. Although we cannot define a precise number of clusters, we can see different regions being formed based on three groups: class 3 (DP-64QAM), class 2 (16-QAM) and a mixture between classes 0 (DP-BPSK) and 1 (DP-QPSK). The set of classes 0 and 1 are physically difficult to classify and they are blurried in most of the regions in the image for no matter what number of features. We also observe that for the top 2 features, it appears to have a direction that change our classification, in which some blurries occur near the boundaries of the supposed decision regions. In these regions, it is hard for a classifier to separate the data. The problem is solved by adding more features. For the top 8 features (which make the classifier have an accu-



(a) 2 features
(b) 4 features
(c) 8 features
(d) 11 features

Figure 4: Data visualization using PCA for classes DP-BPSK (black), DP-QPSK (purple), DP-16QAM (blue) and DP-64QAM (light blue)



(a) 2 features
(b) 4 features
(c) 8 features
(d) 11 features

Figure 5: Data visualization using t-SNE for classes DP-BPSK (black), DP-QPSK (purple), DP-16QAM (blue) and DP-64QAM (light blue)

racy greater than 90% as shown before), there are different directions to increase the classification and when the region of 0 and 1 finishes, it usually leads to class 2 and, then, class 3. But using those features we could separate those aforementioned more-specific cases better.

4 Conclusion

In this work, we presented a promising approach to solve modulation format prediction. We measured the performance of different layouts of neural networks, and we could achieve nearly the mathematical performance of optimal decision based on GN approach. When compared to the same procedure found on literature for the same problem, we could observe similar results, attesting the validity of our implementation. Besides that, we used feature selection as a tool to reduce the number of features. Specifically, it would reduce problems for companies to acquire those measurements and, therefore, reducing overall costs of those procedures. Finally, we applied PCA and t-SNE to have different views of the data. For future work, we intend to test other methods in machine learning to keep the same accuracy using less features. Another possible work would be to turn this problem as regression and trying to predict the OSNR value instead of directly classifying the best method to send the electrical signal.

References

Bo, T., Tang, J., and Chan, C. C.-K. (2016). Blind modulation format recognition for software-defined optical networks using image processing techniques. In *Optical Fiber Communications Conference and Exhibition (OFC), 2016*, pages 1–3. IEEE.

Dias, V. O., Pereira, E. d. V., Rocha, H. R., Segatto, M. E., and Silva, J. A. (2017). Performance evaluation of co-ofdm systems based on electrical constant-envelope signals. *Optical Fiber Technology*, 37:30–34.

Guesmi, L. and Menif, M. (2015). Modulation formats recognition technique using artificial neural networks for radio over fiber systems. In *Transparent Optical Networks (ICTON), 2015 17th International Conference on*, pages 1–4. IEEE.

Poggiolini, P. and Jiang, Y. (2017). Recent advances in the modeling of the impact of nonlinear fiber propagation effects on uncompensated coherent transmission systems. *Journal of Lightwave Technology*, 35(3):458–480.

Rafique, D. (2018). Machine learning based optimal modulation format prediction for physical layer network planning. In *2018 20th International Conference on Transparent Optical Networks (ICTON)*, pages 1–4. IEEE.

Ren, H., Yu, J., Wang, Z., Chen, J., and Yu, C. (2017). Modulation format recognition in visible light communications based on higher order statistics. In *Lasers and Electro-Optics Pacific Rim (CLEO-PR), 2017 Conference on*, pages 1–2. IEEE.

Wang, D., Zhang, M., Li, Z., Li, J., Fu, M., Cui, Y., and Chen, X. (2017). Modulation format recognition and osnr estimation using cnn-based deep learning. *IEEE Photonics Technology Letters*, 29(19):1667–1670.