Artificial Intelligence (ECSE 526)

Final Project Report

Blind Digital Modulation Identification Using Neural Networks & Higher Order Statistics

December 2016

Amen Memmi

ID:260755070

amen.memmi@mail.mcgill.ca/amen.memmi@gmail.com

In this term project, we need to work on an idea that embodies some of the concepts we have learned in the AI course. I chose to make use of the approach of neural networks in my field of interest, telecommunication, and that is how started the idea of implementing "Blind Digital Modulation Identification Using Neural Networks & Higher Order Statistics".

Introduction

Blind signals interception has regained more attention recently. An essential step in this process is blindly identifying the modulation scheme. It has its roots in military applications such as communication intelligence and spectrum surveillance. Further, the recent and rapid development in the context of cognitive radio have given modulation identification more prominence in civil applications.

Modulation identification algorithms are generally divided into two categories. The first one is based on decision theoretic approach while the second on pattern recognition. The decision theoretic approach is a probabilistic solution based on a priori knowledge of probability functions and certain hypotheses [1], [2]. On the other hand, the pattern recognition approach is based on extracting some basic characteristics of the received signal called features [3]–[6]. I am interested in the latter approach using neural networks as classifiers of modulation schemes based on higher order statistics HOS as features.

For all simulations in this project, I used Matlab, specifically the communications and artificial neural network toolboxes.

System Model

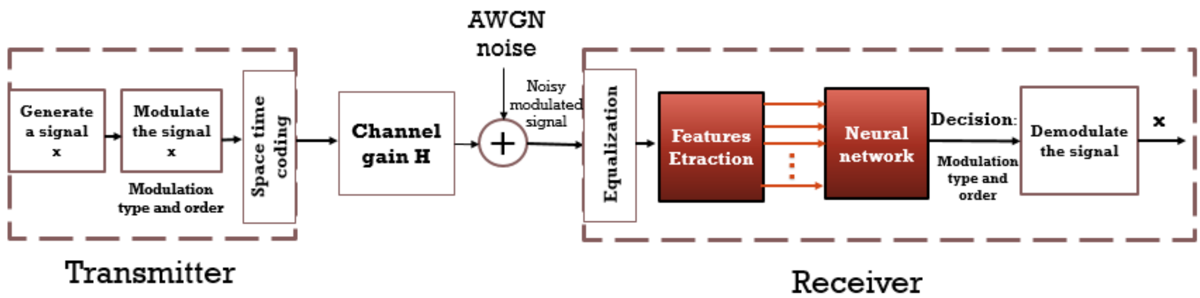


Figure 1 : Scheme of the used communication system

The communication system I am considering consists of a transmitter communicating with a receiver through a noisy channel (additive white Gaussian noise AWGN) as illustrated in the figure 1. Two antennas are used in each side. The transmitter generates its message/ signal, modulates it using one of the digital modulations (PSK, ASK, FSK, QAM…) with a certain order of modulation M, performs space time coding (Alamouti code) and transmits it through the channel. At the reception, equalization is first performed to inverse channel and the HOS are computed to extract the features that are forwarded to the neural network whose task is to decide which modulation has been used allowing the receiver to be able to decode the message.

Features extraction:

One of the important aspects of modulation identification is the selection of the proper identification features. Previous works have shown that higher order cumulants (HOC) and higher order moments (HOM) of the received signal are among the best candidates for signal identification in SISO systems [4], [7]. Higher order moments of a signal x are defined by [18]:

$$M_{km}(x) = E[x^k (x^*)^{k-m}]$$

where k is the moment order. The cumulant of order k of the zero-mean signal x is defined by:

$$C_{km}(x) = Cum\left[\underbrace{x, \ldots, x}_{(k-m)times} \; \underbrace{x^*, \ldots, x^*}_{mtimes}\right]$$

$$\lfloor (k-m)times \quad mtimes \rfloor$$

$$Cum[x, y, z, w] = E(xyzw) - E(xy)E(zw) - E(xz)E(yw) - E(xw)E(yz)$$

For a signal x with N samples, one can estimate the moments as:

$$M_{km}(x) \approx \frac{1}{N} \sum_{p=1}^{n} x^{k-m}(i)\,(x^*(i))^m$$

The features employed in this work consist of a combination of HOM and HOC up to order six. I can note here an important advantage of using the HOS: the complexity of moment computation is of order N (estimating a moment of order k requires only about N complex additions and $k \times N$ complex multiplications). Cumulant calculation is of order N, too. The features extraction process has then a very low complexity $\mathcal{O}(N)$. In table 1, we can find the theoretical values for the used features.

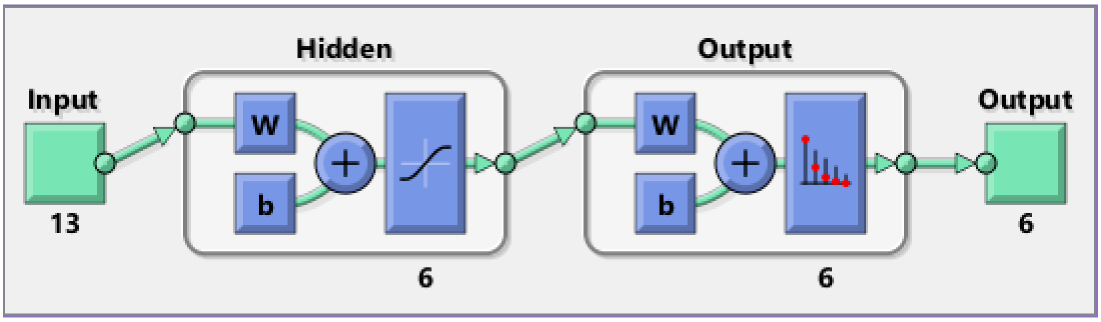
TABLE I

SOME THEORETICAL STATISTICAL MOMENTS AND CUMULANTS VALUES FOR DIFFERENT MODULATION SCHEMES OF INTEREST [1], [5], [7]

	2-PSK	4-PSK	8-PSK	4-ASK	8-ASK	16-QAM	64-QAM
C20	1	0	0	1	1	0	0
M40	1	1	0	1.64	1.77	-0.67	-0.18
M41	1	0	0	1.64	1.77	0	0
M42	1	1	1	1.64	1.77	1.32	1.34
C40	-2	1	0	-1.36	-1.24	-0.68	-0.62
C41	-2	0	0	-1.36	-1.24	0	0
C42	-2	-1	-1	-1.36	-1.24	-0.68	-0.62
M60	1	0	0	2.92	3.62	0	0
M61	1	-1	0	2.92	3.62	-1.32	0.38
M63	1	1	1	2.92	3.62	1.96	2.08
C60	16	0	0	8.32	7.19	0	0
C61	16	-4	0	8.32	7.19	2.08	1.8
C62	16	0	0	8.32	7.19	0	0
C63	16	4	4	8.32	7.19	2.08	1.8

Neural network structure:

After extracting the proper features, the modulation identification problem can be considered as a pattern recognition problem, in which neural networks excel and provide great performance.



The neural network is trained using the resilient backpropagation learning algorithm using the scaled conjugate gradient. Beside the fast convergence (less than 2 seconds/ less than 100 epochs in the following training example), one of its main advantages of lies in the fact that no choice of parameters and initial values is needed at all to obtain optimal or at least nearly optimal convergence times [19]. Also, it is known by its high performance on pattern recognition problems.

The network consists of 13 inputs (features/HOS), 6 hidden layers, 6 output layers and 6 outputs. Outputs are actually the 6 classes/ modulation types I am trying to classify: 2-PSK, 4-PSK, 8-PSK, 8-QAM, 16-QAM and 64-QAM. Optimally then, the output would be a vector containing one and 5 zeros to indicate the chosen class but when the neural network 'is not entirely sure', it can provide probabilities (numbers between 0 and 1) and the decision is made according to the highest value.

The neural network structure including the number of hidden layers has been chosen through simulations. For instance, I did not get better result when considering more hidden layers. This structure is directly related to network training speed and identification precision.

Neural Network training example:

I chose to start by training the neural network at a low noise environment, a Signal to Noise Ratio SNR equal to 15dB (Useful signal ≈ 30 times Noise). The training matrix is a 13 by 6000 matrix: 13 input features * 1000 samples for each of the 6 class. The training data (input and target matrices) is generated through trainingMatricesGeneration.m function or could be imported directly to Matlab Workspace using input_matrices_6000.mat file. Among the 6000 samples, 70% are used for the training while 15% is reserved for each of the validation and testing phases.

The training algorithm using the scaled conjugate gradient converged in less than 2 seconds with an epoch of 93 iterations (it has gone 93 times through the training data to adjust the weights).

The confusion matrices below show that the trained network performed perfectly with a 100% success during training, validation and testing.

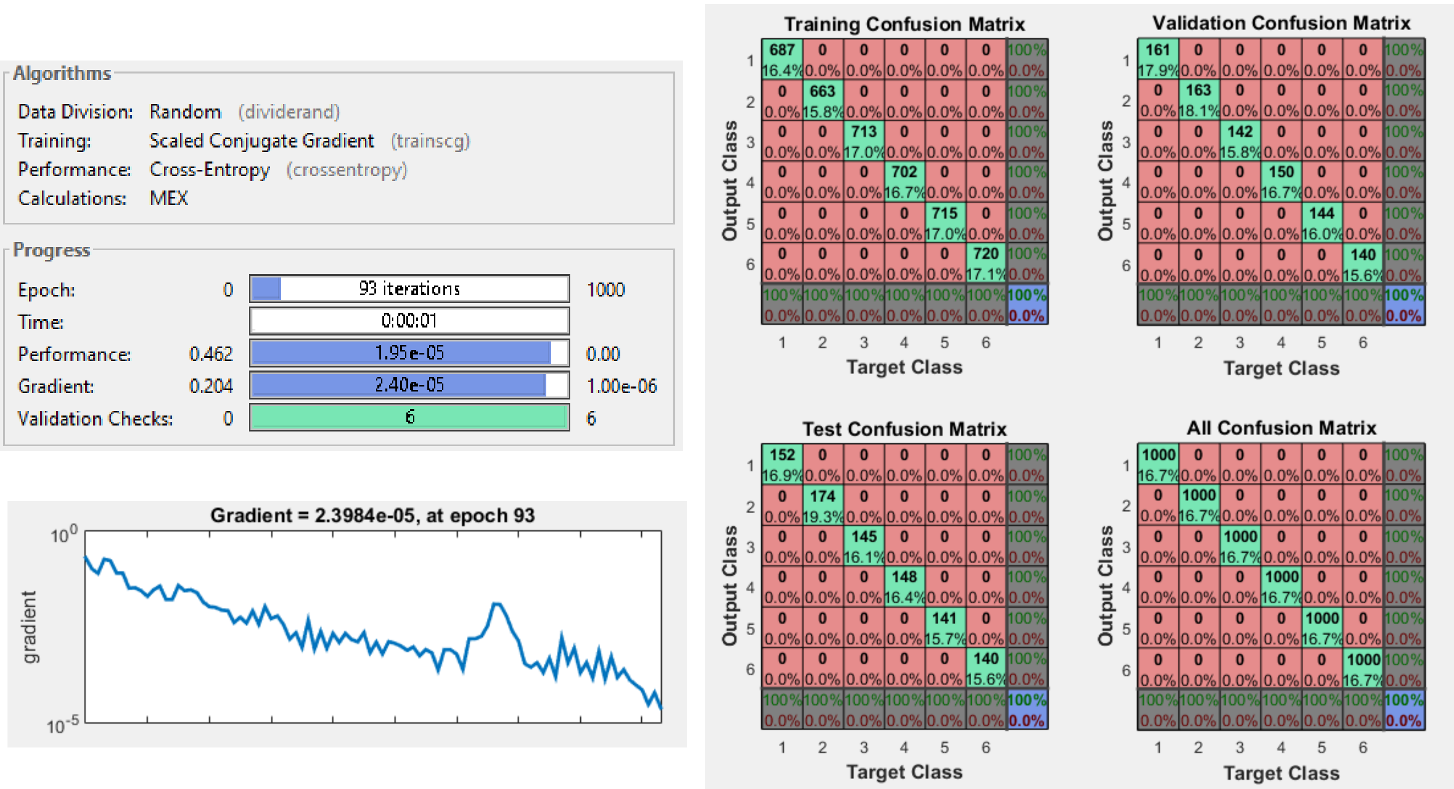


Figure 2: Selected training results

After training, a test phase is launched through a Monte Carlo simulation of $N=10^5$ samples for the same SNR=15 dB, and the neural network is evaluated through the probability of identification as shown in the following table:

Modulation scheme	**2-PSK**	**4-PSK**	**8-PSK**	**8-QAM**	**16-QAM**	**64-QAM**
% of successful identification	100%	100%	100%	100%	98.83%	95.17%

Results show a perfect result for lower modulation order with a small confusion between 16 and 64 QAM but still, 99% overall success rate is observed. However, these results are obtained for the same SNR used at training, which is rarely the case in real world application due to the varying nature of wireless channels. Thus, we should verify performances for different values of SNR. The following plot illustrates this effect.

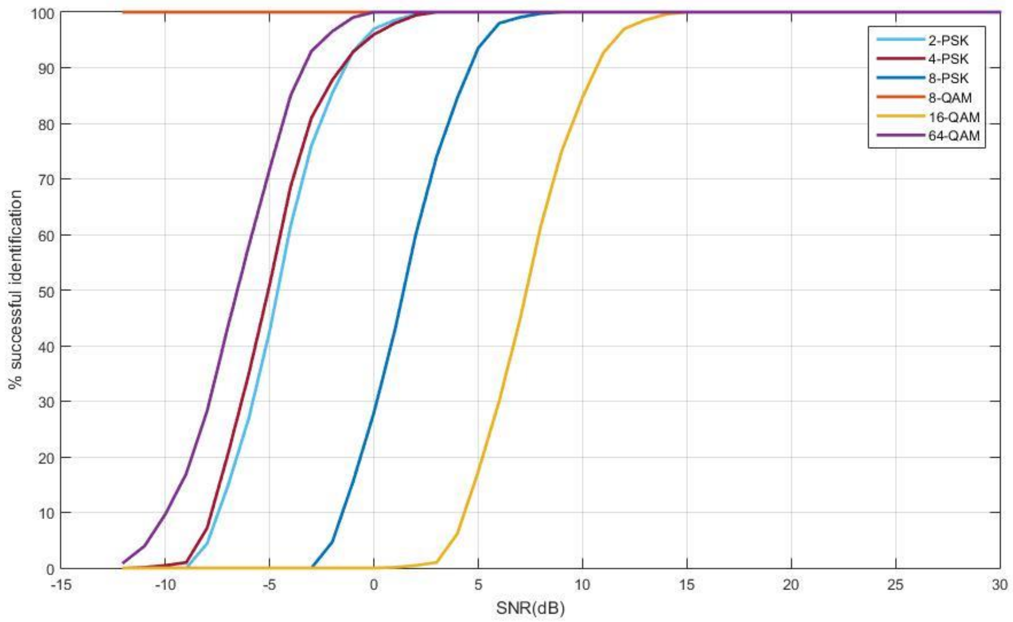


Figure 3: Successful identification as a function of the SNR for the neural network trained at 15 dB

We can see that for lower values of SNR the network looses in performances and stops being able to detect the QAM (and PSK) schemes for value around 0 dB (and 8 dB). The first ideas that comes to mind here is to try training at lower SNR levels. Simulations results are plotted in the following:

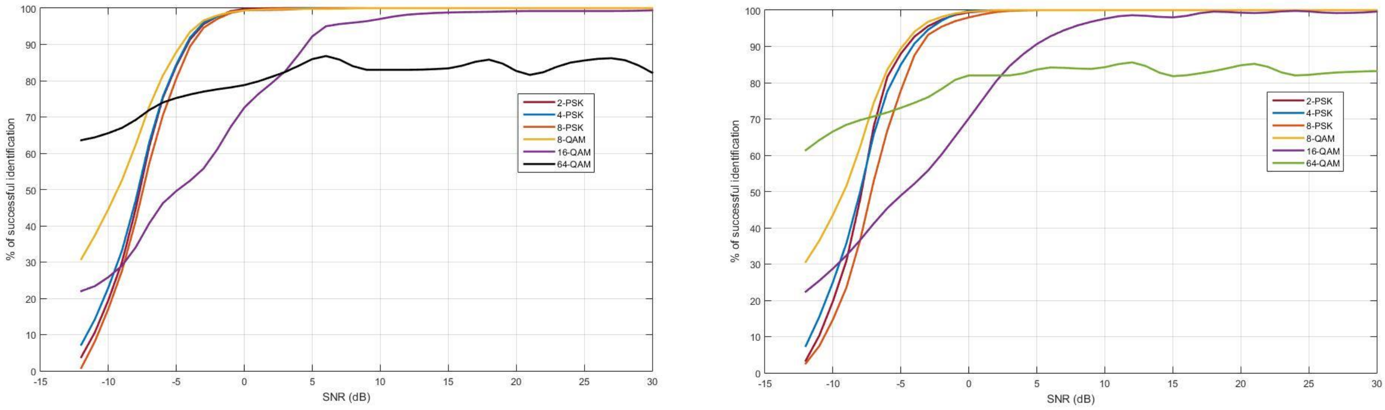
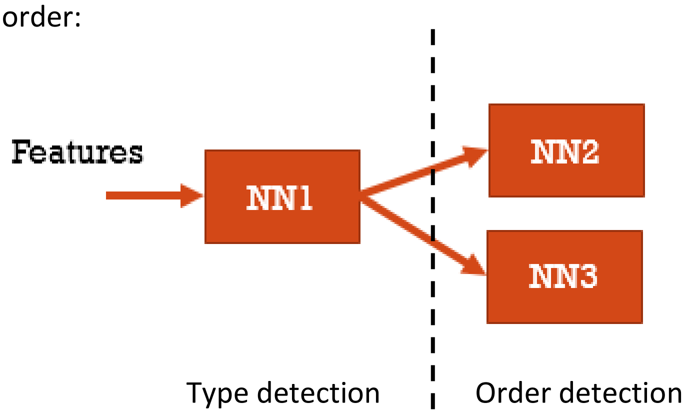


Figure 4 : Successful identification as a function of the SNR for the neural network trained at 3 dB (left) and -9 dB (right)

At lower training SNR, we observe better detection of the QAM schemes but this come at the expense of performance reduction at high SNR regime where we are no more able to be 100% sure. Thus, we could combine the positive of the two previous experiences by considering two neural networks, one trained at high SNR and the other at lower one.

Another interesting idea came to my mind, to use two layer or a cascade of 2 neural networks of different targets/outputs: a first layer to detect the modulation type and a second one to identify modulation order:



I also chose to train the networks at an SNR of 3 dB .The performance of this new configuration is illustrated by the following figure.

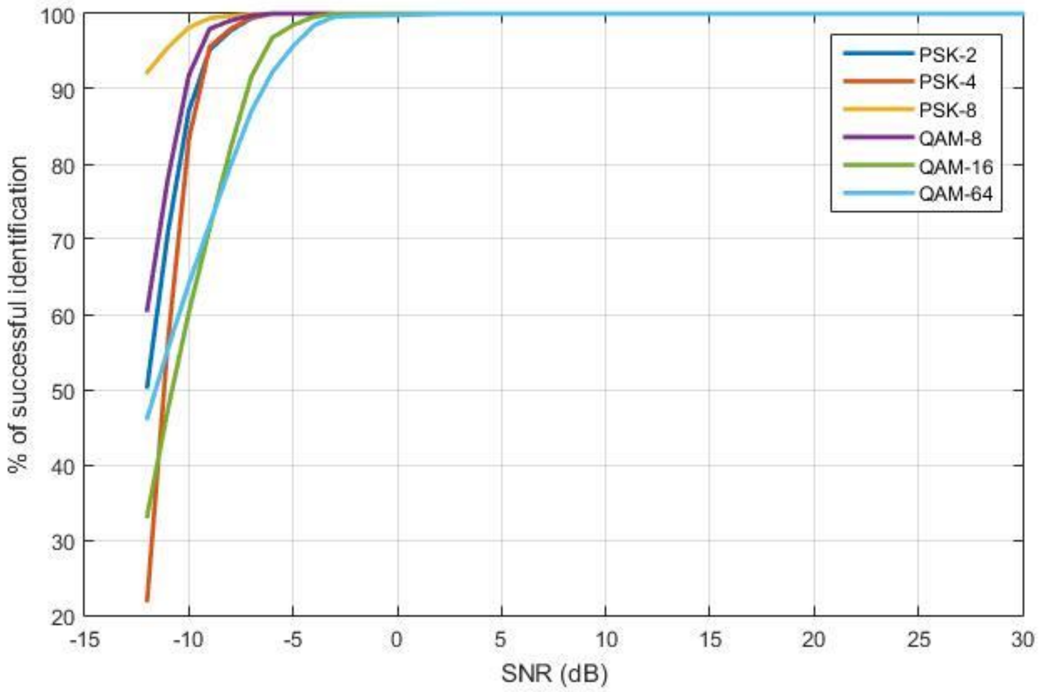


Figure 5: Neural networks cascade trained at 3 dB

This design idea appeared to be efficient performance-wise, since we are able now to identify the modulation scheme and be 100% sure for SNR level of -5 dB or a in the presence of an AWGN noise 3 times more powerful than the signal itself!

Conclusion

This work proves the power of neural networks when it comes to classification. With a training time less than two seconds, a detection time of few milliseconds and mostly a perfect identification success rate for an SNR higher than -5 dB, neural networks present themselves as a strong candidate to be adopted for future application of blind signal identification. This perfect success rate may suffer a performance degradation in real life application, since the channel model I considered here is the simplest and there way more realistic channel model that should be considered to evaluate the real life performance of such classifiers.