How to choose a neural network architecture? - A modulation classification example

Anand N. Warrier, SaiDhiraj Amuru
Department of Electrical Engineering
Indian Institute of Technology - Hyderabad, India
Email: {ee16btech11042, asaidhiraj}@iith.ac.in

***Abstract*— Which neural network architecture should I choose? This is a common question that is heard nowadays. Having searched the slew of papers that have been published over the last few years in the cross domain of machine learning and wireless communications, the authors found that several researchers working in this multi-disciplinary field continue to have the same question. In this regard, we make an attempt to provide a guide for choosing neural networks using an example application from the field of wireless communications, specifically we consider modulation classification. While deep learning was used to address modulation classification quite extensively using real world data, none of these papers give intuition about the neural network architectures that must be chosen to get good classification performance. During our study and experiments, we realized that this simple example with simple channel models can be used as a reference to understand how to choose the appropriate deep learning, specifically neural network, architectures based on the system model for the problem under consideration. In this paper, we provide numerical results to support the intuition that arises for several system models.**

***Index Terms*—Modulation, Classification, Deep Learning, Neural Networks, DNN, CNN, RNN, LSTM, Wireless, Communications, AWGN, Fading, ARMA**

I. Introduction

Machine learning, especially deep learning, has now made in-roads to many fields of engineering, economics, healthcare etc [1]. Likewise, it is now being explored aggressively for a variety of problems in wireless communications [2], [3]. The field of wireless communications was traditionally addressed using a model-based approach as the models of wireless channels, transmission signals, etc. were well studied for a long time using physical experiments and measurements. The big question was whether machine learning and in particular deep learning can add any value to this knowledge.

Some initial studies in the 2016-2017 time frame showed that deep learning could indeed provide significant gains for some problems in wireless communications such as modulation classification [4], end-to-end transmitter-receiver design [5], low complexity channel decoders [6], among others. Some of the initial studies in this field borrowed ideas from the deep learning advancements in the computer vision and natural language/speech processing domains and directly applied some of the well known architectures (such as ResNet [7], Imagenet [8], Densenet [9]) to problems in wireless communications [10]. Owing to this blind hit-and-trial method of application of neural network architectures, there was not much intuition as to why deep learning worked for these wireless communications problems and what kind of deep learning architectures are most suitable for a specific system model. This remains to be a question that is commonly asked even to date despite a slew of works starting from 2016 [4], up until recently in 2020 [11].

In this paper, we address this basic question by considering a case study of modulation classification. Modulation classification is defined as the process of determining the modulation scheme of a noisy signal from a given set of possible schemes. This belongs to a broader set of problems normally termed as signal identification. This classification process has many applications in wireless communications, including in autonomous multi-mode and software-defined radios. Blind modulation classification, i.e., without estimating any parameters for the wireless channel or the underlying signal, has become quite popular especially with the advancement of a variety of classification techniques, especially machine learning techniques [12]. Note that in this paper, we focus on deep learning techniques as they can be used to classify a large set of modulation schemes as discussed in [4], as opposed to simple feature-based traditional modulation classifiers.

While the modulation classification work has been studied using advanced machine learning techniques, especially via deep learning techniques in [4] and [13], there is no intuition provided about the steps and directions to take for choosing a specific deep learning model. The goal of this paper is to answer the most basic questions that upcoming machine learning researchers in the wireless communications field will be able to answer. For this paper, we generated the datasets using Matlab. We can extend this work using the intuitions gained from this paper to any similar datasets available.

A. Experiment setup

The following is the setup for the experiments presented in this paper. We generate several data vectors of in-phase and quadrature component samples (i.e., baseband IQ samples) of length 100 for each modulation class, in this case QPSK, 16-QAM, and 64-QAM. Channel and noise effects are added appropriately to these data vectors to create the received signal vectors of appropriate length (taking the channel length into consideration). We pose this problem as a supervised learning classification task, and therefore we label each of these vectors with the appropriate label corresponding to its modulation class. The training was done at 30dB SNR and testing was done at various SNR values. In all experiments, we started with basic architectures, such as single layer, with 10 nodes, and then started adding more layers and nodes until the training accuracy reached 100%. These architectures will be shown later in Tables I and II. The resulting architectures are used for testing at various other SNR values. For all the results in the paper, we used Keras libraries to implement the various architectures, and Adam algorithm was used for gradient descent optimization.

In what follows, we will use $\mathbf{x}$ to represent the transmit signal which has symbols from a specific modulation class such as QPSK, 16-QAM, 64-QAM; $\mathbf{n}$ represents the additive zero mean-unit variance-complex white Gaussian noise samples; $\mathbf{h}$ is used to represent the complex fading channel coefficients (such as Rayleigh fading) and finally $\mathbf{y}$ is used to represent the received signal. The neural network classifier operates on the received signal $\mathbf{y}$ to identify the modulation scheme. Each sample $x_i \in \mathbf{x}$ belongs to the same modulation class and the classifier must identify the modulation scheme of $\mathbf{x}$ given $\mathbf{y}$. In all this work, we assume perfect synchronization at the baseband level and that the effects of pulse shaping have been removed.

The rest of the paper is organized as follows. In Section II, we first give some intuition about the neural network architectures from a signal processing perspective. Then, in Sections III-V, we discuss the modulation classification problem in 3 different channel models, namely AWGN channel, multipath fading channel, and ARMA-based channel model. In each section, we present the intuitively what networks must be chosen based on the system model and then show numerical results to support these observations. Finally, we conclude the paper in Section VI.

II. Intuition about Neural Network Architectures

In most machine learning works, three architectures are used, fully connected neural network (FCNN) or deep neural network (DNN) or multi-layer perceptron, convolutional neural network (CNN), and recurrent neural network (RNN).

DNN is a feed-forward multi-layer neural network [14] with full connections in all layers. In a DNN, all inputs are treated to be independent of each other. Therefore, DNN architectures are deemed to suit well for the cases when the input data is independent and identically distributed.

CNN is a neural network with convolutional, pooling and fully connected layers [14]. The convolutional and pooling layers act as filters that give a representative feature from the input samples. The operation can be understood as performing an finite-impulse response filtering (FIR) filtering on the input to the neural network where the filter is designed such that the most striking parts of the data can be extracted. For the case of the modulation classification, recollect that we need to recover the input data in its true form to be able to classify. Hence, if a CNN must be used for modulation classification, it must be able to learn an FIR filter that operates on $\mathbf{y}$ to recover $\mathbf{x}$. The activation functions add non-linearity to the system, therefore a CNN can be understood as an FIR filter with some non-linearity.

The recurrent neural network architectures are ones where connections between nodes form a directed graph. This allows them to exhibit a temporal mapping over the inputs. This is especially useful when past information can be used to get a better understanding of the current information. These networks can be understood as implementation of an infinite impulse response filter (IIR) wherein the output of an IIR filter depends not only on the inputs, but also on past outputs. The activation functions at various stages in the network add non-linearity to the system. Therefore, a RNN can be understood as an IIR filter with some non-linearity. In this paper, we implement the RNN using a popular architecture, namely "Long short-term memory" network aka LSTM [14].

III. AWGN Channel

In this section, we will discuss how to choose a neural network for modulation classification when the following system model is used

$$\mathbf{y} = \mathbf{x} + \sqrt{\frac{1}{\mathtt{SNR}}}\mathbf{n}, \tag{1}$$

where $\mathtt{SNR}$ is the signal-to-noise-ratio in linear scale. In this model, each sample $y_i \in \mathbf{y}, i = 1, 2, \ldots, \mathtt{length}(\mathbf{y})$ is independent and identically distributed as each sample n_i is i.i.d complex Gaussian distributed. Therefore, there is no dependency among the samples of y_i. In this case, as per intuition, we can use every sample as a feature point and be able to identify the modulation scheme of the incoming signal. Based on this reasoning, a DNN-based architecture seems to be enough for classifying the modulations. We now have to see if this intuition indeed translates to performance. In this paper, when we employ a DNN, the input dimension is 2 wherein a single real value and the corresponding imaginary value is fed as input to the network. DNNs consider the input nodes independently. So, we split each 100×2 samples to 100 data points of pairs of real and imaginary samples. For finding the modulation scheme, the DNN gives us a modulation scheme for each (I,Q) sample. We use the majority rule to decide the modulation scheme of the entire 100×2 sample.

Apart from DNN, the various architectures we used for the training and testing over the given dataset for understanding the network architectures necessary for this system model,

Architecture	Layers	Nodes per Layer	Input Dimensions	Output Dimensions
DNN	4 layers	128,64,32,3	2	3
CNN	2 convolution layers, 1 flattening, 2 dense	64 filters (3*1), 16 filters (3*2),128,3	100*2	3
RNN	2 LSTM layers, 2 fully connected	128,128,32,3	100*2	3

TABLE I: Various neural network architectures used for modulation classification in AWGN channel

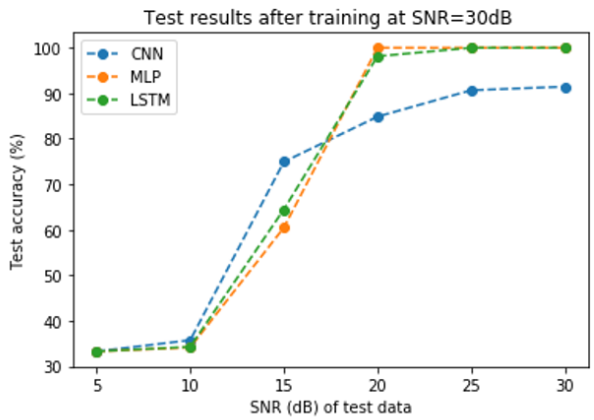

Fig. 1: Modulation classification performance of architectures in Table I in AWGN channel.

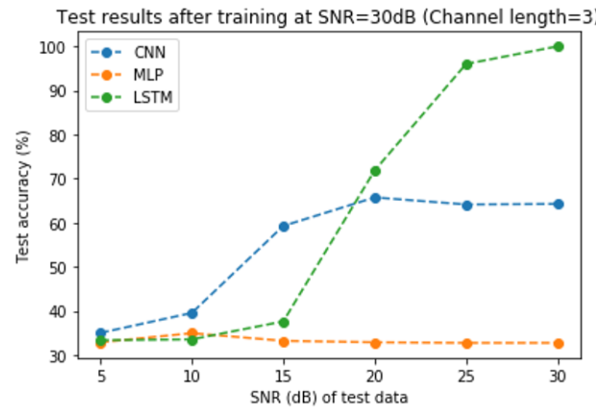

Fig. 2: Modulation classification performance of architectures in Table I for system model (2) with $L = 3$, as a function of SNR.

are shown in Table I. The performance is shown in Fig. 1. From these results, it is seen that all architectures more or less perform the same. We observed that as we increase the hyper parameters (such as number of filters in case of CNN, or number of layers) in each architecture, they will all perform the same. This is because the data is quite benign and i.i.d. Based on the intuition given earlier, it is clear that a DNN is sufficient for good modulation classification performance in the AWGN channel owing to the fact that there is no correlation among the various samples in the input data vector. Therefore a complicated convolution architecture, or recurrent neural network architecture is not required for problems when it is known that the samples are i.i.d.

IV. Multipath Fading Channel

In this section, we will discuss how to choose a neural network for modulation classification when the system model is

$$\mathbf{y} = \mathtt{conv}(\mathbf{h}, \mathbf{x}) + \sqrt{\frac{1}{\mathsf{SNR}}}\mathbf{n}, \tag{2}$$

wherein $\mathtt{conv}(\mathbf{h}, \mathbf{x})$ indicates the convolution operation between $\mathbf{h}$ and $\mathbf{x}$. $\mathbf{h}$ is Rayleigh fading channel with L-taps where L is unknown and $E\left(|\mathbf{h}|^2\right) = 1$. The classification is done without the knowledge of the wireless channel which is where the training procedure is useful. We assume for this work that the channel remains the same across the training and testing datasets as the goal is to identify the type of neural-networks that must be used with system model as (2).

Consider an example noiseless case with $L = 2$. Then we have,

$$[y_1, y_2, \ldots] = [h_1 x_1, h_1 x_2 + h_2 x_1, \ldots], \tag{3}$$

$$[x_1, x_2, \ldots] = \left[\frac{y_1}{h_1}, \frac{y_2}{h_1} - \frac{y_1 h_2}{h_1^2}\right]. \tag{4}$$

The convolution operation is similar to performing a FIR-filtering operation on the data $\mathbf{x}$ using the filter $\mathbf{h}$. In order to perform modulation classification, we need to recover the true data $\mathbf{x}$. From the above equations, it is obvious that to get $\mathbf{x}$, we need to perform an IIR-filtering type operation using the base channel coefficients $\mathbf{h} = [h_1, h_2, \ldots]$. So, intuitively, we need to perform the inverse operation of FIR filtering i.e., IIR filtering to recover $\mathbf{x}$ from $\mathbf{y}$ (of course the effect of noise still remains). For this operation, looking at the available neural network architectures, as per discussion in Section II the recurrent neural network architecture fits well.

The details of the neural networks used for this system model are shown in Table. I.

The performance of the various neural networks architectures as a function of SNR and channel length L is shown in Figs. 2, 3. From these results, it is clearly seen that LSTM-based RNN architectures performed well for the considered multipath fading channel models. In our experiments, we observed that DNN architecture performed well only when $L = 2$ and for cases with $L > 2$, the DNN architecture did not perform well as they cannot capture the correlation in the data vector $\mathbf{y}$ induced by the channel. Similarly, CNN-based architectures which are similar to an FIR filter could not remove the correlation (created by the FIR filtering operation

Architecture	Layers	Nodes per Layer	Input Dimensions	Output Dimensions
DNN	4 layers	128,64,32,3	2	3
CNN	2 convolution layers, 1 flattening, 3 dense	64 filters (3*1), 16 filters (3*2),128,64,3	100*2	3
RNN	2 LSTM layers, 2 fully connected	128,128,32,3	100*2	3
CLDNN	1 1D convolutional layer, 1 LSTM layer, 4 dense	64 filters (filter size=3),128,64,32,3	100*2	3

TABLE II: Various neural network architectures used for modulation classification in ARMA channel.

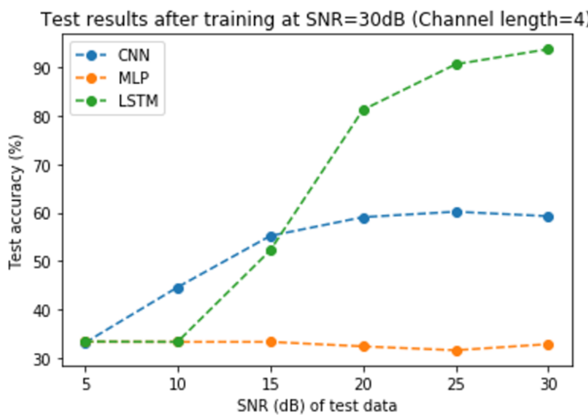

Fig. 3: Modulation classification performance of architectures in Table I for system model (2) with $L = 4$, as a function of SNR.

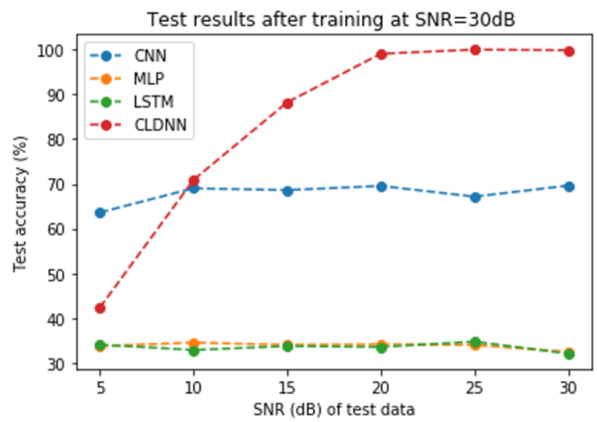

Fig. 4: Modulation classification performance of architectures in Table II for system model (5) with $L = 1$, $K = 3$, as a function of SNR.

of the wireless channel) among the various samples to recover $\mathbf{x}$ which explains their poor performance. We also found that CNN architectures were sensitive to the channel length and their architecture had to be changed according to the channel length as the filters and the number of layers in the network must be capable of learning the correlation in the dataset. However, the LSTM architecture worked irrespective of the channel length as it can very well utilize its reverse side connections, clearly indicating that LSTM (RNN) must be used for this problem. The LSTM-based RNN networks performed an IIR-filter type operation (as explained in Section II) which removed the FIR filtering operation of the wireless channel and therefore performed quite well. Note that as part of the LSTM network implementations, we ensured that the gradients did not explode by saturating the gradients [14].

V. ARMA Channel

In this section, we will discuss how to choose a neural network architecture when the following system model is used-

$$y_i = \sum_{l=0}^{L-1} h_l x_{i-l} + \sum_{k=0}^{K-1} g_k y_{i-k-1} + \sqrt{\frac{1}{\mathsf{SNR}}}\mathbf{n}, \qquad (5)$$

where $\mathbf{h} = [h_0, h_1, \ldots, h_{L-1}]$ and $\mathbf{g} = [g_0, g_1, \ldots, g_{K-1}]$ are L-length and K-length complex vectors that represent the moving average (MA) and auto-regressive (AR) process parameters respectively. ARMA models are typically used to model non-linearity introduced by amplifiers present in the transmit-receive chain or by the wireless channel itself [15]. See that when $K = 0$, then the model degenerates to the one studied in Section IV. So from now, in this section we assume that $K \geq 1$. The architectures used for the training and testing over the given dataset are shown in Table II. For this section, we studied a new architecture termed as CLDNN and will be explained shortly why this new architecture is used.

From (5), it is seen that two operations contribute to (5); (a) an FIR filtering operation of the input signal $\mathbf{x}$ and an IIR filtering operation on the past data samples of $\mathbf{y}$. If only the FIR filtering operation is present, then as done in previous section only LSTM-based network would have been sufficient to recover the data $\mathbf{x}$. And if only IIR filtering operation is performed, then a CNN-based network would have worked well. However, in this case, both the operations contribute to (5). Hence, for this problem we used a new architecture termed as CLDNN, which has a combination of convolutional, LSTM and dense layers to account for both the operations performed in (5). The Table II shows the various architectures tried for this model.

The performance of the various architectures is shown in Fig. 4, 5. The results clearly show that CLDNN architecture significantly performs well compared to the architectures used earlier. And as per the intuition given earlier, the mixture of convolutional layers and LSTM layers together helped for the improved performance. It is observed that for the case when $L = 1, K = 3$, the CLDNN architecture performs quite well, but the performance of the same architecture drops to 67% when $L = 2$ and $K = 3$. These experiments inform us that a neural network architecture must be carefully chosen based on the system model under consideration.

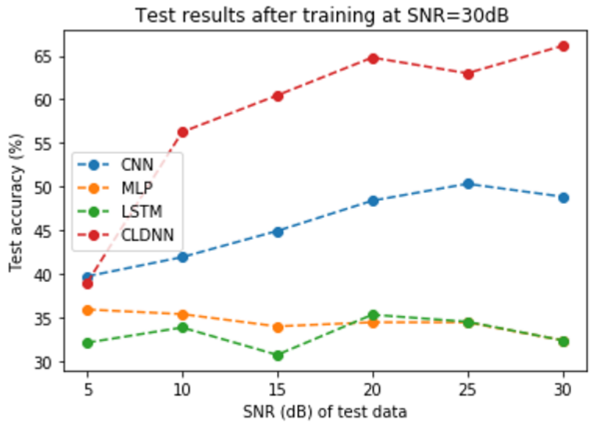

Fig. 5: Modulation classification performance of architectures in Table II for system model (5) with $L = 2$, $K = 3$, as a function of SNR.

VI. Conclusion

In this paper, we considered the example of modulation classification in the field of wireless communications and studied in detail how a machine learning model must be chosen, specifically we studied the case of choosing the most appropriate neural network architectures based on the system model. We showed that in AWGN-type channels, DNN-based architectures perform well. For the case of multipath fading channels i.e., for channels modeled as a convolution of a fading channel and the transmit signal (akin to moving-average-type MA model), RNN-based architectures performed well as they performed an inverse operation of the channel convolution with the data signal. For the case of ARMA-type fading channels, depending on the length of the AR and MA processes (or filters), the required architecture changed. Since these models have impact of both the AR and MA models, we used a mixture of CNN and RNN networks to create a new architecture called as CLDNN which performed quite well. From all these observations, it is clear that the type of the neural network architecture to be used depends on the underlying system model and a researcher working in this domain must make the appropriate choices rather than relying on a blind hit-and-trial method.

References

[1] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature,* vol. 521, no. 7553, pp. 436–444, May 2015.

[2] C. Zhang, P. Patras, and H. Haddadi, "Deep Learning in Mobile and Wireless Networking: A Survey," in *IEEE Commun. Surv. Tut.,* vol. 21, no. 3, pp. 2224-2287, Mar. 2019.

[3] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based AI-based or both?" *arXiv preprint*, arXiv:1902.02647 Feb. 2019.

[4] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks", in *Proc. Int. Conf. Eng. Appl. Neural Netw.* pp. 213-226 2016.

[5] T. J. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognitive Commun. Netw.,* vol. 3, no. 4, pp. 563–575, 2017.

[6] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *Proc. Allerton Conf. Commun., Control, Computing* (Allerton), Monticello, IL, pp. 341-346, Oct. 2016.

[7] K. He, *et. al.*, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.* (CVPR), pp. 770–778, Jun. 2016.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv.Neural Inf. Proc. Syst.* (NIPS), Dec. 2012.

[9] G. Huang, *et. al.,* "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.* (CVPR), pp. 2261-2269, Jun. 2017.

[10] X. Liu, D. Yang, and A. El Gamal, "Deep neural network architectures for modulation classification," in *Proc. IEEE Asilomar Conf. Signals Syst. Comput.* (ACSSC), pp. 915-919 Oct. 2017.

[11] E. Bjornson and P. Giselsson, "Two Applications of Deep Learning in the Physical Layer of Communication Systems," *arXiv preprint*, arXiv:2001.03350, Jan. 2020.

[12] S. Amuru and C. R. C. M. da Silva, "A Blind Pre-Processor for Modulation Classification Applications in Frequency-Selective Non-Gaussian Channels," in *IEEE Trans. Commun.,* vol. 63, no. 1, pp. 156-169, Jan. 2015.

[13] S. Ramjee, *et. al.*, "Fast deep learning for automatic modulation classification," *arXiv preprint*, arXiv:1901.05850, Jan. 2019.

[14] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning." MIT Press, 2016.

[15] G. B. Giannakis and E. Serpedin, "Blind identification of ARMA channels with periodically modulated inputs," in *IEEE Trans. Signal Process.*, vol. 46, no. 11, pp. 3099-3104, Nov. 1998.