DEEP LEARNING-BASED DECODING FOR PHASE SHIFT KEYING

*Mete Yildirim, Radosveta Sokullu**

Ege University, Department of Electrical and Electronics Engineering
Turkey

* Corresponding Author, e-mail: radosveta.sokullu@ege.edu.tr

Abstract: In this paper, the application of Deep Learning (DL) in the field of telecommunications is discussed, focusing specifically on symbol detection at the receiver. The performance of Deep Learning-based detection is examined for phase shift keying modulation over Additive White Gaussian Noise (AWGN) and Rayleigh channels. First, a model is proposed which shows that the theoretical bit error rate and throughput can be achieved using DL techniques. Then, the effects of different DL model parameters on the model performance are investigated. The DL model for symbol detection with tuned and minimized parameter set is examined from various aspects and it is shown that this improved version can achieve the desired results with much less complexity of the realization.

Key words: AWGN, Deep Learning, Detection, Machine Learning, PSK.

1. INTRODUCTION

In recent years DL has become one of the most popular and efficient machine learning methods [1], [2]. Details on DL can be found in [3], written by the pioneers of the field. Considering its performance and continuously extending possibilities, it can be stated without exaggeration that, in the near future, it will take its place as one of the major driving technologies in many areas of our lives. Resources point out that it will have the effect of the industrial revolution that happened a long time ago [3]. DL can be applied in various fields to solve complex problems that are very hard to model mathematically. In most recent years, it has also proved to be very useful for communication systems (CS) engineering applications. Even though there are a number of applications related to the higher levels of the OSI reference model, DL has not yet manifested its full potential in issues related to the physical (PHY) layer of electronic CS [4–7]. According to [8], DL has no commercialized application for the PHY layer. A reason for this might be the historically motivated resistive behavior of the telecommunication industry. Other reasons include cost considerations in adopting a new technology but the most important reason comes from the fact that there exist no telecommunication specific DL models developed in an optimized manner addressing the issues at the PHY layer. This has motivated the work in this paper aiming to research DL application in the detection process and propose a specific optimized DL model for signal detection in modern CS.

2. RELATED WORK

The literature survey indicates that even though the use of Neural Networks (NN) in communication systems is quite old [9], it does not exactly include DL specifically, a major player that has changed machine learning (ML) drastically nowadays becoming one of its major pillars. Hence, most of the recent studies focus especially on the use of DL in various aspects and layers of CS design and operation. DL can be used for coding-decoding, channel estimation and equalization, resource allocation, signal and modulation recognition, end-to-end communication system, demodulation, etc. In [8], an autoencoder is proposed that combines the functions of all units of a traditional communication system in one entity and shows that the autoencoder can achieve the coded-BPSK (Binary Phase Shift Keying) throughput in an end-to-end manner with better BER(Bit Error Rate) throughput in the higher signal to noise ratio (SNR) range. In [10], the autoencoder concept is applied to a MIMO system that introduces a new paradigm in terms of optimization, combining many important communication blocks such as coding, decoding and feed-back in one entity. An example of a practical implementation of the autoencoder is provided in [11]. Article [12] examines symbol detection in MIMO systems with DL and claims nearly optimal detection in real time. To reduce the complexity and to handle both detection and channel coding, [13] offers a new DL model that delivers remarkable performance improvement for decoding compared to conventional MIMO. More references related to the application of DL for different functional layers can be found in [14–20]. However, these works do not consider signal detection for individual modulation schemes and their performance related to the specific DL system parameters. This work considers the limited, even though very important case for PSK, used in many existing communications systems especially for control signaling (i.e BPSK). The main contributions are:

- To show that using DL symbol detection can achieve the max theoretical throughput under AWGN and Rayleigh channels.
- To examine DL model parameters in detail and study their effects on the system performance in order to obtain a practically optimized DL- based model for symbol decoding.

From here on the paper is organized as follows: first the basics of DL are briefly reviewed and the proposed DL model is discussed followed by receiver throughput estimation for AWGN and Rayleigh fading channels and comparison to the theoretically achievable throughput in sec. 3. Sec. 4 examines the DL model parameters and their effects on BER. Finally, the optimized model with the least complexity is experimentally obtained and it is shown that it can achieve acceptable BER performance with much reduced complexity.

3. DEEP LEARNING BASED DETECTION

3.1. Deep learning foundation

A NN is composed of layers and neurons. A neuron does a simple calculation on given input features by multiplying them with arbitrary generated very small numbers. The final output of a neuron is estimated with an activation function $g(w^T x + b)$, where w and b are weight vector and bias respectively. In order to achieve the power of non-linear output estimation, the activation function, $g(.)$, is generally non-linear, as the examples given in Table.1. The stack of neurons considered column-wise is called a layer. When the number of layers is increased the NN structure becomes a Deep Neural Network (DNN). The process of training such a network is known as Deep Learning. The learning may be supervised or unsupervised. In this study only supervised learning is considered, for which the NN model is provided the input features and ground truth output for those features. Except the input features ***x*** and the output ***y***, the middle features are unknown and set by the network itself which is known as end-to-end learning. The model must be provided with enough training examples to complete the learning, i.e $(x^{((i))}, y^{((i))})$. It allows finding the best multiplying factors to the input features (vector) to calculate the desired output ***y***. The computation of a neuron for the input features vector ***x*** is given as $z = w^T x + b$, where ***w*** is a weighing vector and ***b*** is constant, hence a layer computation is given as $\boldsymbol{Z} = \boldsymbol{W}^T x + \boldsymbol{b}$ where ***W*** is a matrix, ***b*** is a vector. ***Z*** is a vector transformed by the activation function and applied to the next layer. ***W*** and ***b*** are randomly initialized to small numbers, using the normal distribution. The last layer may have one or more neurons and for binary classification it has a single neuron. The NN is realizing the mapping of $\boldsymbol{N_i}$ dimensions of the input to $\boldsymbol{N_0}$ dimensions of output given as $f(x;W,b):R^{Ni} \to R^{No}$ in general. The calculated output is used in a cost function which can be maximized by using different algorithms. For example, in binary classification the logistic regression log likelihood given in Eq. (1) is used.

$$\sum_{i=1}^{m} \left(y^{(i)} \log a^{[L](i)} + (1 - y^{(i)}) \log(1 - a^{[L](i)})\right) \tag{1}$$

In Eq. (1), *m* and *L* represent the number of training examples and the layers respectively. More on Deep Learning, including multinomial classification can be found in [3].

Table 1. The most common activation functions.

sigmoid	$g(z) = 1/(1+e^{(-z)})$
relu	$g(z) = \max(z, 0)$
tanh	$g(z) = (e^z - e^{-z})/(e^z + e^{-z})$

3.2. Symbol Detection

One of the most popular digital modulations is binary phase shift keying (BPSK) in which data is conveyed by changing the phase of a constant frequency of the carrier wave. It is also known as 2 -PSK. Digital information in the form of zeros and ones is generally realized using $s_0(t) = -\sqrt{2E_b/T_b}\cos(2\pi ft)$ and $s_1(t) = \sqrt{2E_b/T_b}\cos(2\pi ft)$ for 0 and 1 respectively. BPSK signals are one-dimensional and can be represented in one dimensional signal space with $s_0 = -\sqrt{E_b}$, $s_1 = \sqrt{E_b}$. The received signal is given as $y = s + n = \sqrt{E_b} + n$ where *n* stands for the AWGN noise component with zero mean and variance $\sigma^2 = (1/2)N_0$. If $y > 0$, the receiver decides that a "***1***" is transmitted and if $y < 0$, it decides that a "***0***" is transmitted. Given that the probability density function of noise is $p(x) = (1/(2\pi\sigma^2))\exp((-x-\mu)/2\sigma^2)$, the probability of error when s_1 is transmitted is $p(e|s_1) = (1/2)erfc(\sqrt{E_b/N_o})$, and the probability of error when s_0 is transmitted is $p(e|s_0) = (1/2)erfc(\sqrt{E_b/N_o})$. Assuming that s_1 and s_1 are equally likely probable, the total probability of error is $P_b = (1/2)erfc(\sqrt{E_b/N_o})$. In the case of Rayleigh channel, the received symbol is $y = hs + n$, where h is a Rayleigh random variable and has probability density function given as $p(h) = (h/\sigma^2)\exp(-h^2/2\sigma^2)$. At the receiver-end the estimation is done based on $\tilde{y} = y/h = (hx + n)/h = x + \tilde{n}$. The bit error rate (BER) is changed due to the change in the effective bit-energy-to-noise ratio. The instant bit-energy-to-noise ratio is given as $(|h|^2E_b)/N_o$ and the BER becomes $P_{b|h} = (1/2)erfc(\sqrt{\gamma})$, where $\gamma = (|h|^2E_b)/N_o$ for instant values of *h*. The average BER is calculated by evaluating it over the pdf of γ that has a chi-square distribution given as $p(\gamma) = (E_b/N_o)^{-1}\exp(\bar{\gamma}\gamma_o/(E_b/N_o)), \gamma \geq 0$. The average BER is calculated using Eq. 2:

$$P_b = \int_0^\infty (1/2)erfc(\sqrt{\gamma})p(\gamma)d\gamma = \frac{1}{2}\left(1 - \sqrt{(E_b/N_o)/((E_b/N_o)+1)}\right) \qquad (2)$$

The 4-PSK is a two-dimensional signal. Considering that $\{\mp 1 \mp 1j\}$ is the alphabet used for symbol creation in the given signal space, the transmitted symbols may be given as $s_0 = \sqrt{E_a/2}(1+1j), s_1 = \sqrt{E_a/2}(-1+1j), s_2 = \sqrt{E_a/2}(-1-1j)$ and $s_3 = \sqrt{E_a/2}(1-1j)$, where $\sqrt{E_a/2}$ is the energy normalizing factor. The received symbol is given as $y = s_n + n$, where *n* is AWGN noise component with zero mean and variance equal to $\sigma^2 = (1/2)N_0$. The conditional probability distribution function (pdf) of *y*, given s_n is transmitted, is $p(y\,|\,s_n) = 1/\sqrt{\pi N_o}\exp\left(-\left(y - \sqrt{E_a/2}\right)/N_o\right)$. The error probability for each symbol can be given in terms of receiving a symbol correctly as $P_{QPSK} = 1 - p(c\,|\,s_n)$. Considering that all the symbols are equally likely transmitted, receiving a symbol correctly is given as $p(c\,|\,s_n)$. For example, the reception of symbol s_2 correctly is given as $p(c\,|\,s_2) = p(\Re y > 0\,|\,s_2)p(\Im y > 0\,|\,s_2)$, where $\Re, \Im$ stands for the real and imaginary part respectively. $p(\Re y > 0 \mid s_2)$, $p(\Im y > 0 \mid s_2)$ can be given as:

$$p(\Re y > 0 \mid s_2) = 1 - \frac{1}{\sqrt{\pi N_0}} \int_{-\infty}^{0} e^{\frac{-\left(\Re y - \sqrt{\frac{E_s}{2}}\right)^2}{N_0}} dy \quad (3)$$

and

$$p(\Im y > 0 \mid s_2) = 1 - \frac{1}{\sqrt{\pi N_0}} \int_{-\infty}^{0} e^{\frac{-\left(\Im y - \sqrt{\frac{E_s}{2}}\right)^2}{N_0}} dy \quad (4)$$

Solving, $P_{QPSK} = 1 - p(c \mid s_n)$ the probability of error can be found as:

$$P_{QPSK} = erfc\left(\sqrt{\frac{E_s}{2N_0}}\right) - \frac{1}{4} erfc^2\left(\sqrt{\frac{Es}{2N_0}}\right) \quad (5)$$

4. MODEL IMPLEMENTATION

4.1. Implementation Procedure

In this section, the proposed model and its implementation in DL is presented. First the procedure is outlined in general, followed by the details of the proposed solution. The steps in creating and training a DL model are summarized in Fig.1:

1. Loading the input data: The data is stored in a 2D-array where 1st and 2nd dimensions represent rows and columns respectively. The first two columns, holding the received data, are separated to be used as input to the model and the last column, holding the corresponding ground-truth symbols is used as output.

2. Definition of the model: In the second step, a sequential model is created by adding fully connected layers one after other, making sure the input layer has two inputs while the output has two and four outputs depending on the type of modulation used. The number of neurons, the activation functions, and the drop-out factor (used to deactivate some neurons randomly to overcome overfitting) are set.

3. Compiling of the model: In this step, the weights, appropriate lost argument (i.e. binary cross entropy etc.) and optimizer are chosen. The optimizer, "*adam*", a popular version of gradient descent, has been selected as it automatically tunes itself and gives good results in a wide range of problems, as explained in [21].

4. Fitting of the model: The execution of the model for a certain number of epochs and batches that can be later set up to get the model to converge.

5. Evaluation of the model: The performance of the model is tested on a new dataset and loss and accuracy of the model are calculated.

6. Prediction solution. When a reasonable accuracy is obtained, the weights of the model are kept to be used in the prediction of the symbols.

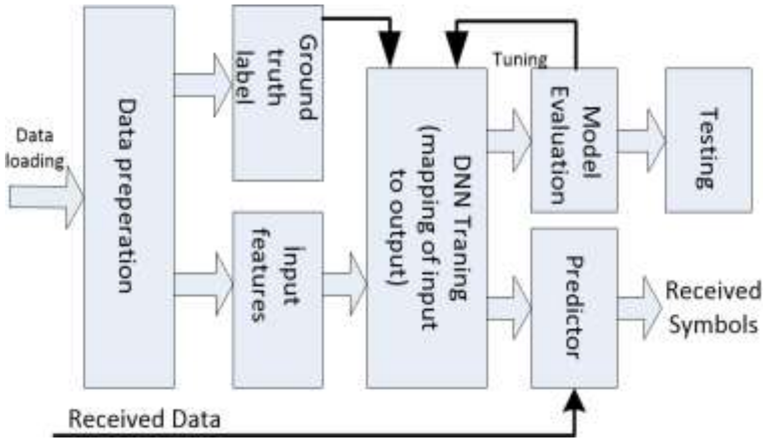

Figure 1. Model training, testing and prediction

For this study, the general steps provided above are customized and the labeled data is created first. Secondly, the various models created are trained. Finally the trained models are tested on different data not seen by the model in training. The case of BPSK detection, where the received symbols can take only two different values, is considered as a binary classification problem. The transmitted symbols are "***1***"and "***-1***" for bit "***1***" and "***0***" respectively. The received symbols can take any value on a two-dimensional vector space due to the additive Gaussian noise.

First, we show that an arbitrary model can achieve the theoretical upper bound of bit error rate given in Eq. (5). The results of this experiment are given in Fig.2 compared with the theoretically calculated throughput. They clearly show that the obtained BER for BPSK and 4-PSK, assuming the channel is either AWGN or Rayleigh, with the proposed non-optimized DL model exactly match the theoretical bit error rate for both channels.

4.2. Optimizing the model

The second goal of this work is to optimize the proposed model. The aim of the optimization is to create a model that achieves high throughput in terms of BER with less complexity of the NN structure. This, in return, reduces hardware cost, training time and warranty normal-fitted model (not over nor under trained). In general, a NN depends on a large number of parameters that need to be tuned for reducing the cost and complexity if possible, without sacrificing the system performance. In this section the performance of NN with respect to some parameters of the model will be examined in detail. The following parameters are taken into consideration: activation function, number of layers, number of neurons per a layer, and neuron distribution to the hidden layers.

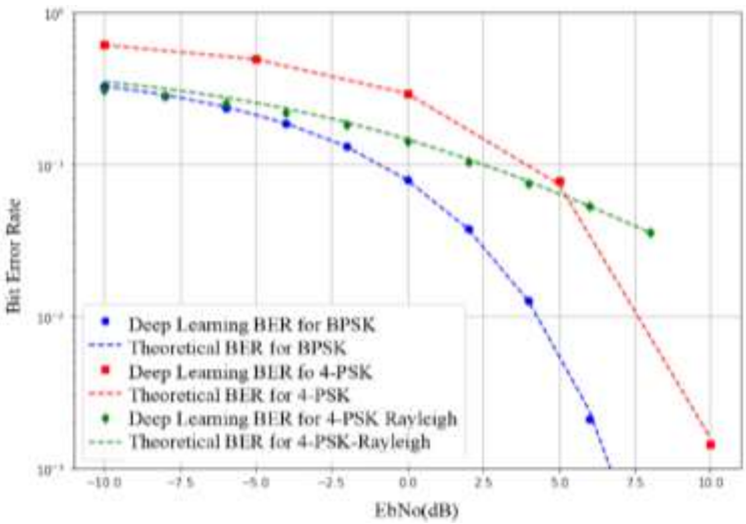

Figure 2. BER of DL detection in comparison to the ideal throughput

First, the effect of the number of neurons in the hidden layers on the output performance are examined. The high number of neurons is a parameter that should be treated with caution as it extends the training period and can cause over-fitting. The results of the simulations are presented in Fig.3. It can be seen that, when there are 512 neurons in the hidden layers, the model achieves the exact performance of the ideal throughput. When the number of neurons is reduced to 256, the model still produces a very good result, but when the number is reduced further to 128 neurons, the model shows an acceptable performance only for 4 -PSK. As a second important parameter the distribution of neurons in the different layers is examined. Simulations were carried out with the following distributions: (1). 48 neurons as 2,40,6 for 1st, 2nd, 3rd layers respectively, (2). 48 neurons as 8, 8, 32 for 1st, 2nd, 3rd layers respectively and etc. Other distributions and their throughput in terms of the bit error rate are shown in Fig.4. It can be clearly seen that the best performance can be achieved with a uniform distribution of neurons into the hidden layers.

One of the important hyper-parameters of the DL model is the number of epochs considered. An epoch is defined as the time for which an entire dataset is passed both forward and backward through the network only once. Fig.5 summarizes the results of how the number of epochs affects the bit error rate for the case of 4-PSK. On purpose, a simple model is chosen to keep the training process faster. The number of epochs is increased without putting the model under over-training. As can be seen in Fig.5, increasing the number of epochs produces better bit error rate results. By evaluating the results, it can be advised that to stay within a well-fitting learning regime, 64 epochs should be used.

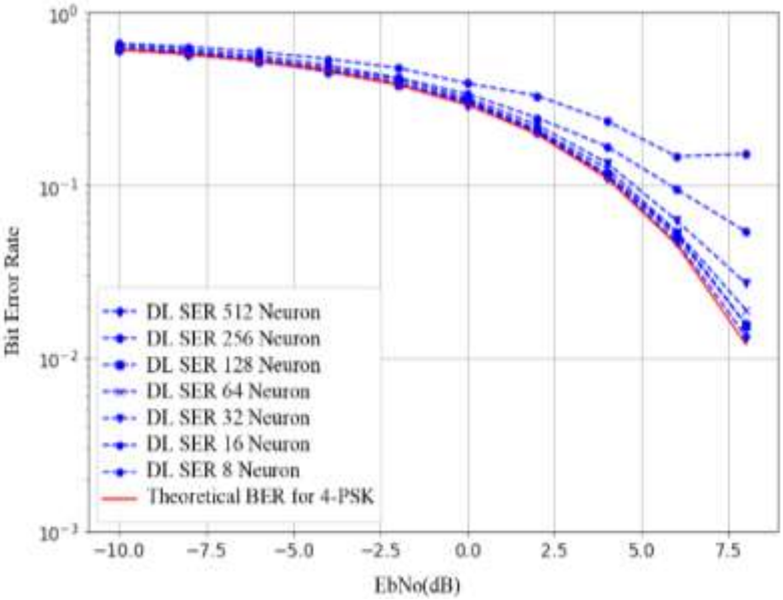

Figure 3. Effects of increasing number of neurons in the hidden layers on the BER

It should be noted also that the same throughput can be achieved by increasing the number of neurons in the hidden layers which requires much less training time. This can be explained by the fact that back-propagation is a much more time-consuming process than forward-propagation.

5. DISCUSSION

In this work, a DL network model is proposed for solving the symbol detection problem for the 2-PSK and 4-PSK case. The extensive simulations carried out suggest that neural networks can handle the digital symbol detection quite well. To reach the best possible symbol detection results for PSK, further tuning of the model parameters is proposed and evaluation is provided considering minimizing the model parameters to avoid waste of resources and reduce training time. Table 2 shows that there exists a big gap between minimized and non-minimized models in terms of the total number of parameters used for symbol detection for BPSK demodulation. This gap gets even larger for PSK-4. We can naturally expect that a higher modulation order will exhibit a greater gap. The tuned and optimized models also show that they can easily achieve the ideal bit error rate throughput in the considered channels (AWGN, Rayleigh). The optimized model given in Table 2 is the suggested model for BPSK. In the case of 4 -PSK, the number of neurons should be set to 64.

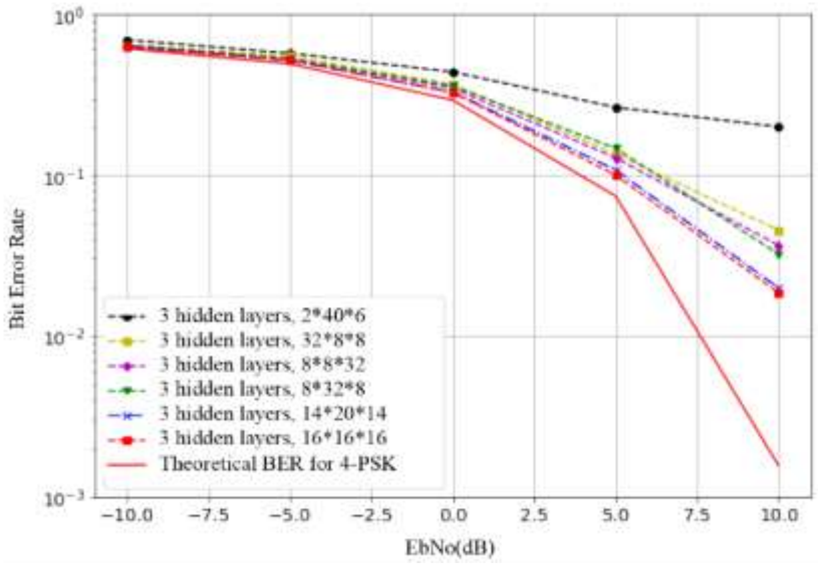

Figure 4. Effect of the distribution of the neurons on the BER

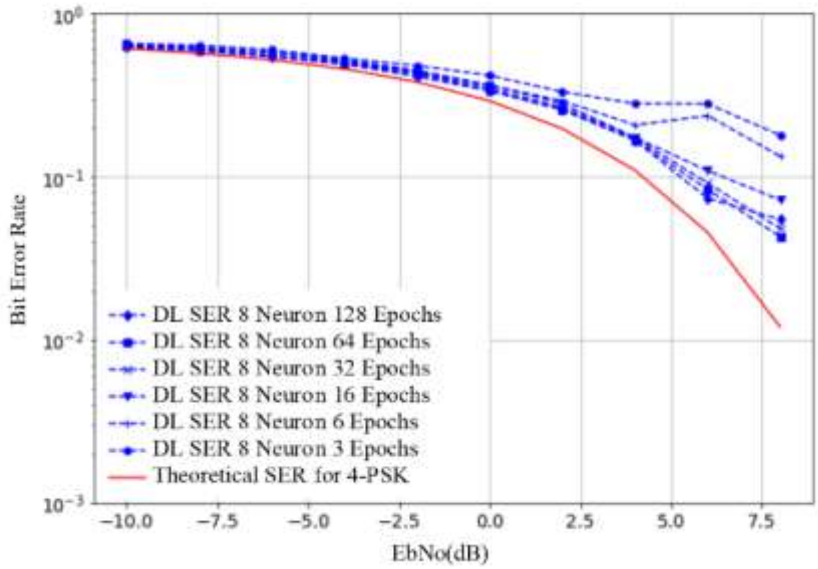

Figure 5. Effects of reducing the number of epochs on the BER for 4-PSK

Table 2. Optimized vs Non-optimized model for BPSK

Non-optimized model			***Optimized model***		
Layer	Output Shape	# of parameter	Layer	Output Shape	# of parameter
Dense	(None, 512)	1536	*Dense*	(None, 16)	48
Dropout	(None, 512)	0	*Dropout*	(None, 16)	0
Dense	(None, 512)	262656	*Dense*	(None, 16)	272
Dropout	(None, 512)	0	*Dropout*	(None, 16)	0
Dense	(None, 512)	262656	*Dense*	(None, 16)	272
Dropout	(None, 512)	0	*Dropout*	(None, 16)	0
Dense	(None, 512)	262656	*Dense*	(None, 16)	272
Dropout	(None, 512)	0	*Dropout*	(None, 16)	0
Dense	(None, 2)	1026	*Dense*	(None, 2)	34
	# of total param.	790,530		# of total param.	898

The following concrete observations and suggestions for simulations are made:

a) The more neurons used in hidden layers; the better results are obtained.

b) In order to achieve acceptable performance while reducing the complexity a minimum of 64 neurons per layer is required for the case of 4-PSK.

c) If the number of neurons is further reduced, the same performance can be achieved by increasing the number of epochs leading to increased training time.

d) Uniform distribution of neurons for the hidden layers achieves better throughput so the detection model should have even distribution of neurons.

e) Better performance for higher modulation order is obtained by increasing the number of hidden layers and it also requires more training data.

Our simulation results show that finetuning of the designed DL model is a very important issue, which is also confirmed [11]. DL in general allows creating models for CS that can learn to communicate over any type of channel and prior mathematical model is not required. This is also true for the case at hand. The assumption for AWGN channel can easily be extended to cover other channel models.

6. CONCLUSIONS

This paper describes two possible applications of Deep Learning methods in the area of telecommunications related to symbol detection at the receiver side. The limited, though very important case for PSK, which is still used in many existing communications systems especially for control signaling (i.e BPSK) is covered. First a short summary of the basic concepts and operations involved in the Deep Learning paradigm is presented, followed by details on the implementation of Deep Learning methods for signal detection in the case of BPSK and PSK modulation over AWGN and Rayleigh channels. It is shown that these methods can achieve error performance close to the theoretical limits. Furthermore, specific parameters of the DL system and their effects on the accuracy and bit error rate are evaluated. Finally, a DL model with minimized parameter set is proposed which is proved to achieve the desired results with much less complexity. In our future works the proposed model will be extended to cover other transmission schemes including more complex encoding and decoding approaches.

REFERENCES

[1] Y. LeCun, Y. Bengio, G. Hinton. Deep learning. *Nature*, ISSN: 0028-0836, vol. 521, no. 7553, 2015, p. 436, https://doi.org/10.1038/nature14539

[2] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar. A Survey on Deep Learning: Algorithms, Techniques, and Applications. *ACM Computing Surveys.*, ISSN: 03600300, vol. 51, no. 5, 2018, pp. 92:1–92:36, https://doi.org/10.1145/3234150

[3] I. Goodfellow, Y. Bengio, A. Courville. *Deep Learning*. ISBN: 9780262035613, MIT Press, US, 2016

[4] C. Zhang, P. Patras, and H. Haddadi. Deep Learning in Mobile and Wireless Networking: A Survey. *IEEE Communications Surveys Tutorials*, 2019.

[5] Z. Qin, H. Ye, G. Y. Li, and B.-H. F. Juang. Deep learning in physical layer communications. *arXiv preprint, 2018, arXiv:1807.11713*.

[6] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin. Deep learning for wireless physical layer: Opportunities and challenges. *China Communications*, vol. 14, no. 11, 2017, pp. 92–111.

[7] H. Wang, Z. Wu, S. Ma, S. Lu, H. Zhang, G. Ding, and S. Li. Deep Learning for Signal Demodulation in Physical Layer Wireless Communications. *IEEE Access*, vol. 7, 2019, pp. 30792–30801, doi:10.1109/ACCESS.2019.2903130

[8] T. J. O'Shea and J. Hoydis. An introduction to machine learning communications systems, *arXiv preprint*, 2017, vol. 1702.

[9] M. Ibnkahla, Applications of neural networks to digital communications–a survey. *Signal processing*, vol. 80, no. 7, 2000, pp. 1185–1215.

[10] T. J. O'Shea, T. Erpek, and T. C. Clancy. Deep Learning Based MIMO Communications. arXiv, 2017, 1707.07980v1.

[11] S. Dörner, S. Cammerer, J. Hoydis, and S. ten Brink. Deep learning based communication over the air. *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, 2018, pp. 132–143.

[12] N. Samuel, T. Diskin and A. Wiesel. Deep MIMO detection. International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Sapporo, 2017, pp. 1-5.

[13] T. Wang, L. Zhang, and S. C. Liew. Deep Learning for Joint MIMO Detection and Channel Decoding. arXiv, 2019, 1901.05647v1

[14] Q. Chen, S. Zhang, S. Xu and S. Cao. Efficient MIMO Detection with Imperfect Channel Knowledge - A Deep Learning Approach. *Wireless Communications and Networking Conference (WCNC)*, Marrakesh, Morocco, pp. 1-6, 2019, doi: 10.1109/WCNC.2019.8885582.

[15] S. Takabe, M. Imanishi, T. Wadayama and K. Hayashi. Deep Learning-Aided Projected Gradient Detector for Massive Overloaded MIMO Channels. I*nternational Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1-6, arXiv:1806.10827

[16] N. Farsad and A. Goldsmith. Neural network detection of data sequences in communication systems. *arXiv preprint*, *2018*, *arXiv:1802.02046*.

[17] V. Corlay, J. J. Boutros, P. Ciblat, and L. Brunel. Multilevel MIMO Detection with Deep Learning. *Asilomar Conference on Signals*, *Systems, and Computers*, 2018, pp. 1805–1809.

[18] A. Al-Baidhani and H. H. Fan. Learning for Detection: A Deep Learning Wireless Communication Receiver Over Rayleigh Fading Channels. *International Conference on Computing, Networking and Communications (ICNC)*, 2019, pp. 6–10, doi: 10.1109/ICCNC.2019.8685517.

[19] H. Ye and G. Y. Li. Initial Results on Deep Learning for Joint Channel Equalization and Decoding. *Vehicular Technology Conference (VTC-Fall)*, 2017,Canada, pp. 1–5, doi: 10.1109/VTCFall.2017.8288419

[20] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, 2015, pp. 1137–1149,

[21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint, 2014, arXiv:1412.6980*.

Information about the authors:

Mete Yıldırım received his B.S. from Marmara University and M.S. and Ph.D. degrees in electrical and electronic engineering from Fatih University. He completed his post-doc studies in the Department of Electrical and Electronics Engineering, Ege University. His current research interests include wireless communication, cooperative communication, D2D communication, machine learning and deep learning

Radosveta Sokullu received her PhD in Telecommunications from the Technical University Sofia and joined the Dept. of EEE at Ege University as an Assistant Professor in 2000. She became Head of Telecommunications Branch in 2004, Associate Professor at the same department in 2016 and Head of the Wireless Communications Lab. Her current research covers different wireless communication networks and protocols (IEEE 802.15.4, Bluetooth, Wireless Sensor Networks, Cellular Networks) with a focus on MAC layer and PHY layer protocol design, resource sharing, resource allocation, IoT and Machine-to-Machine Communications. Most recently she has focused on cell free MIMO, massive IoT networks and incorporating machine learning methods for resource allocation and random access in 5G and Beyond 5G networks.

Manuscript received on 17 December 2020