Journal of Engineering Science and Technology
Vol. 17, No. 4 (2022) 2776 - 2787

DECODING OF TURBO CODE AND POLAR CODE USING DEEP LEARNING FOR VISIBLE LIGHT COMMUNICATION

ALDRIN CLAYTUS VAZ[1], C. GURUDAS NAYAK[1,*], DAYANANDA NAYAK[1], NAVYA THIRUMALESHWAR HEGDE[2]

[1]Department of Instrumentation & Control Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal-576104, Karnataka, India
[2]Department of Aeronautical & Automobile Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal-576104 Karnataka, India

Abstract

The success of neural network-based systems in areas such as image classification, computer vision, pattern recognition, Natural language Processing etc. has made its usage also expand in the area of Wireless communication. This paper investigates the decoding of two codes widely used in modern communication viz, Turbo Codes and Polar Codes using Deep Learning (DL) methods. The aim of this study is to explore the feasibility of using DL architectures based on Deep Neural Networks (DNN) and Recurrent Neural Networks (RNN) for decoding of Polar Codes and Turbo Codes, respectively. The decoding performance of DNN based Polar Codes is also investigated based on number of neurons in each layer, activation functions and number of layers. Simulation is carried out in MATLAB for the communication system independently for the above codes over an Additive White Gaussian Noise-Visible Light Communication Channel employing Colour Intensity Modulation. The results compare the performance of the traditional decoding algorithm with the proposed DL approach over different Signal to Noise Ratio (SNR) regimes. The RNN based Turbo decoder outperforms the conventional Turbo decoder in terms of Bit Error Rate performance at lower SNRs and the DNN based Polar decoder has a similar performance as of the conventional Polar decoder.

Keywords: Deep neural networks, Forward error correction, Gated recurrent unit, Polar code, Recurrent neural networks, Turbo code, Visible light communication.

1. Introduction

Modern day Indoor illumination is revolutionized by solid-state lighting. Light Emitting Diodes (LEDs) are replacing incandescent and fluorescent lamps at a rapid pace. The advantages of LEDs are that they are extremely high energy efficient, greater durability, produce less heat and reproduces colour more realistically. LEDs have the capacity of changing to varying light intensities at a high rate, which has led to the development of a new technology for communication called Visible Light Communication (VLC), where LED sources can be used for communication [1]. VLC can be considered as an alternate and a preferred communication technology because of the availability of large bandwidth and immune towards electromagnetic interference from other electromagnetic sources [2]. With the growth and technical development of LED technology, interests on VLC systems have increased concurrently. A LED can provide an energy efficient illumination, simultaneously providing a very high modulation bandwidth for communication. The LEDs are widely used in street lighting, indoor lighting, automobiles etc. which can be used as a source of communication. Moreover, the development in the area of solid-state lighting has further led to the substitution of florescent lamps by LEDs, which inspires the usage of VLC.

The evolution of 5G communication from 4G requires various factors to be taken into consideration such as achieve high data rates, support diverse Quality of Service and accurate processing requirements etc., which makes the need for advanced communication algorithms [3-5]. Due to these demands, researchers are focussing on the use of Deep Learning (DL) methods and have achieved substantial improvements. Various DL architectures such as Generative Adversarial Networks (GANs), Recurrent Neural Networks (RNNs), Convolutional Neural networks (CNNs), Deep Belief Networks (DBNs) are being applied in areas of Drug design, Natural language processing, Computer Vision etc. and have obtained good results [6].

The application of Artificial Neural Networks (ANNs) in the decoding of various Forward Error Correction (FEC) Codes can be found in the literature. The initial works can be found in [7-9] where ANNs are applied for decoding Block Codes and Hamming Codes. In [10, 11], decoding of Convolutional codes was performed using ANNs. Similarly, works on decoding Turbo Code [12], Linear Density Parity Check Code [13] and Polar Code [5] using ANNs can be found.

An approach to decode Turbo Codes based on Feedforward neural network structure was given in [14], which reduced the errors during Turbo decoding operations. CNNs based Turbo Auto encoder is proposed in [15], achieving good performance in low to middle Signal-to-Noise Ratio (SNR) region. A RNN based Turbo decoder is proposed in [16] to implement the decoding operations for various code rates, obtained a Bit Error Rate (BER) of the order 10^{-5} to 10^{-6}. A RNN based Turbo encoding and decoding operation is carried out for Quadrature Phase Shift Keying (QPSK) and Quadrature Amplitude Modulation (QAM) in [17]. The proposed method obtained good performance at lower SNRs. A sparse neural network based on Belief Propagation (BP) is used for decoding Polar codes in [18]. The results showed 60% reduction in complexity, achieved 0.5 dB gain and reduced latency over conventional polar decoding. In [19], Deep Neural Network (DNN) is used for (64, 32) and (512, 256) polar codes. The proposed method also reduces hardware cost and complexity by about 50%. A CNN-BP based Polar decoding gave better performance in terms of BER and gain compared to conventional

method [20]. An unsupervised neural network Polar decoder based on Syndrome loss is proposed in [21], which gave on par performance as of conventional decoding. In [22], BP algorithm and Successive Cancellation (SC) algorithm is implemented using neural networks. The simulation showed the same results as of conventional decoding performances.

DL based approaches are finding success in various fields of engineering/non-engineering domains. Due to the large data handling capabilities of ANNs, an approach is attempted to decode two of the most popular FEC codes (Turbo and Polar Code) used in the modern 4G/5G communication technology. The training of such network architectures may be highly computationally complex. However, the field applications can be achieved at a lower complexity for such trained networks. Considering the level of Performance and complexity, it can be concluded that the deep learning is a competitive decoding method.

The proposed work investigates the application of DL architecture in the decoding of Turbo Code and Polar Code performed at the PHY layer of the communication. The Long-Term Evolution (LTE) variant of the Turbo Encoder/Decoder is chosen [23]. The decoding procedure is framed as supervised learning and the decoding is performed using a RNN architecture. As the size of the DNNs increases, the optimization of its weights also becomes more complex due to increase in complexity of its architecture. Hence, a simple DNN is chosen. A (16, 8) Polar Code is selected, and the decoding is performed using Feed-forward DNNs. BER criteria is used to compare the performance over a range of SNR to the standard LTE decoder and SC with List decoding [24] for Turbo Code and Polar Code, respectively.

The work is organised as follows: An introduction to the encoding/decoding operations of the LTE Turbo Code and (N, K) Polar Encoder/Decoder is presented in Section 2. Section 3 gives the DL approach for Turbo Decoder using RNN. Section 4 explains the DL architecture for Polar Code. Section 5 concludes the paper with scope for future work.

2. Design of Turbo Encoder/Decoder for LTE and (N, K) Polar Encoder/Decoder

2.1. Turbo Code

Turbo codes are formed by concatenating two convolutional codes in parallel form [25]. Two identical Recursive Systematic Convolutional (RSC) component codes are parallel concatenated to get the turbo encoder structure. The inputs given to these two RSC encoders are separated by an interleaver. The role of interleaver is to permute the data given to it in a predetermined way. Both the RSC encoders output their systematic bits and parity bits. The systematic bits of first RSC encoder and the parity bits of both the encoders are used to construct the output of the Turbo Code. As the systematic bits of second RSC encoder is the permuted version of the systematic output bits of first RSC encoder, it is not considered in the final output of the Turbo Code. The idea of parallel concatenation leads to the idea of iterative feedback decoding which is advantageous over serial concatenation as it improves the overall performance of the system. The factors such as number of decoding iterations, constraint length of the encoders, interleaver types, decoding algorithms, and generator polynomials affect the performance of the Turbo Code.

Figure 1 shows the component diagram of a Turbo Code which uses two identical RSC component encoders. X_k and $Z_k$are the systematic and parity bits, respectively of the first RSC encoder and Z'_k is the parity bits of second RSC encoder. Therefore, for every input bit, the output of the Turbo Code produces three output bits thus making its code rate as 1/3. For the same sequence, the two RSC encoders will either produce a low weight code word or a high weight code word [26].

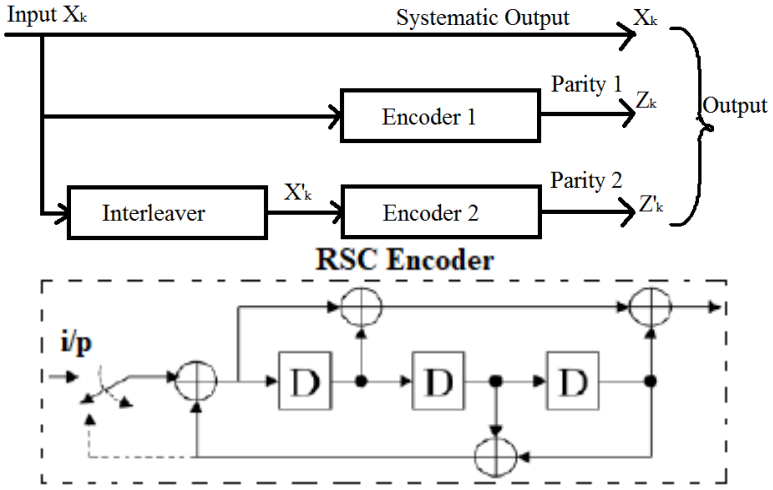

Fig. 1. LTE turbo encoder of code rate 1/3.

Each of the constituent RSC encoder has the transfer function given by Eq. (1) as

$$G(D) = \left[1, \frac{g_1(D)}{g_o(D)}\right] \tag{1}$$

where $g_o(D) = 1 + D^2 + D^3$ and $g_1(D) = 1 + D + D^3$. The state of each encoder is 8 having a constraint length of 4 [27].

When the transmitted code bits (X_k , Z_k and Z'_k) gets passed via a communication channel, they get corrupted and are received at the input of a Turbo decoder as X_{ck}, $Z_{ck}$and Z'_{ck} which is shown in Fig. 2. Hence, due to the addition of channel noise in the received values, they differ from that of the transmitted values. The Turbo decoder estimates the original transmitted information by using a decoding algorithm via a number of iterative decoding steps.

The Turbo decoder is build using two 'Soft-in-Soft-out' (SISO) component decoders to obtain the iterative decoding. The inputs and outputs of each SISO decoders are the a-priori and a-posteriori information, respectively. As seen in Fig. 2, the decoding strategy is that the information is iteratively exchanged among the SISO decoders. The process of exchanging the information is repeated until the output converges to a threshold value. Finally, both the SISO decoder outputs are combined, and final output is obtained by hard decision [26]. The Log-likelihood ratios (LLR's) are used as the soft values of the output which represents the estimated bit as '0' or '1' depending on the LLR to be negative/positive value, respectively, and is calculated as given in Eq. (2).

$$L_R = ln\left(\frac{P[m_k=1|\hat{y}]}{P[m_k=0|\hat{y}}\right) \tag{2}$$

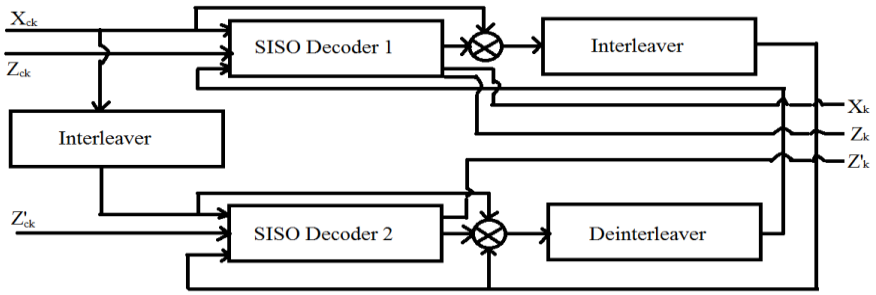

Fig. 2. Iterative turbo decoder.

2.2. Polar Code

A Polar code, shown in Fig. 3, falls into a category of a linear block code, represented by (N, K), where N denotes the codeword length and $K$is the message bits. $G_N = B_N F^{\otimes n}$ is the generator matrix, where the matrix B_N represents the bit-reversal permutation matrix and $F^{\otimes n}$ is the n^{th} order Kronecker product of F, given by Eq. (3) as [28],

$$F = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \quad (3)$$

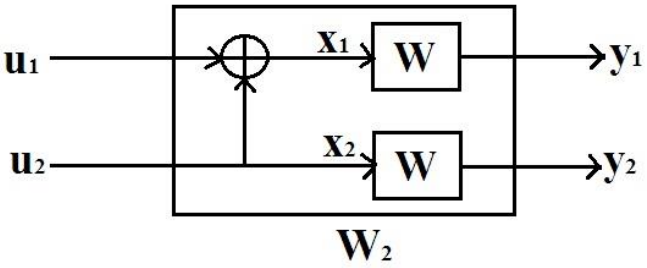

Fig. 3. N=2 Polar encoder.

Let the output of the encoder is a vector x_1^N of length N, computed as $x_1^N = u_1^N G_N$, where $u_1^N = u_1, \ldots, u_N$ is the data sequence. The data sequence can be split into two parts i) the ones whose indices corresponds to the information bits and are to be transmitted on the reliable channels (Set U); and ii) the remaining indices which corresponds to the frozen bits are to be transmitted on the unreliable channels (Set U^c). Hence, we have $x_1^N = u_A G_N(U) \oplus u_{A^c} G_N(U^c)$. The mutual information represents the capacities of the reliable channels. The information bits are the ones with high capacities and the remaining are the frozen bits whose values are set to zero. In the simulation, we have considered N=16, K=8 and Code rate R=1/2.

The SC decoder proposed by Arikan's is the fundamental polar decoder which achieves capacity with moderate complexity. The decoded sequence $\hat{u}_1^N$ is successively estimated by the SC decoder from the received sequence $\hat{y}_1^N$, corresponding to the input sequence u_1^N. N decisions are taken by the decoder for each u_i [29]. The decoder sets $\hat{u}_i = 0$ if u_i is a frozen bit and if $u_i$is an information bit, the LLR is calculated by estimating all previous bits u_1^{i-1} as given in Eq. (4) as,

$$L_N^i\left(y_1^N, \hat{u}_1^{i-1}\right) = \frac{W_N^i\left(y_1^N, \hat{u}_1^{i-1}|0\right)}{W_N^i\left(y_1^N, \hat{u}_1^{i-1}|1\right)} \quad (4)$$

3. DL design for LTE Turbo Code

The design and analysis of ANN based turbo decoder using DL is investigated. The LTE variant of Turbo encoder and Turbo decoder are used. The encoding is done via rate 1/3 Turbo encoder and decoding is carried as a supervised learning operation. RNN architecture is proposed to decode the data and the BER performance is compared with the standard LTE turbo decoding block.

A particular task 'T' is executed by a DL algorithm based on its learning experience 'E' by optimising a particular performance metric 'M'. They can be categorised depending on the labelling of input dataset samples as Supervised and Unsupervised. These algorithms belonging to ANNs class where a number of hidden stacked layers with arbitrary number of neurons are present. The weighted inputs add a neuron along with its biases and passed through a nonlinear activation function and the result passes into the successive layers. Mapping is performed at each layer i, with n_i inputs and m_i outputs as $f^{(i)}: R^{n_i} \rightarrow R^{m_i}$. By taking the input of the ANN as r_i and r_o as output, the mapping of input-output data can be defined by successive nonlinear activations based on their weights and biases (θ) as given in Eq. (5)

$$r_o = f(r_i; \theta) = f^{(L-1)}\left(f^{(L-2)}\left(\ldots\left(f^{(0)}(r_i)\right)\right)\right) \tag{5}$$

where L represents the number of ANN layers.

During the training process of ANN, in order to obtain optimal weights a particular loss function is defined which needs to be optimised. Usually, the loss function is an error function which needs to be minimised over the training set in order to obtain the final weights by means of backpropagation algorithm and gradient descent optimization methods. The activation function considered is a sigmoid function given by $\frac{1}{1+e^{-x}}$. Adaptive Moment Estimation optimizer is used, and the Cost function 'C' considered is given in Eq. (6),

$$C = -\big(ylog(p)\big) + (1-y)\log(1-p) \tag{6}$$

Out of many available RNNs, Gated Recurrent Unit (GRU) is selected which uses reset gates and update gates. There is no memory unit present in GRU and exposures the concealed contents without any control. The reason of using GRU units is because of low computational complexity and good performance. Due to all these advantages, GRUs is selected as the core units for the RNN structure proposed.

The RNN chosen is a bidirectional GRU having activation given in Eq. (7) as

$$h_t = z_t \tilde{h}_t + (1-z_t)h_{t-1} \tag{7}$$

where $\tilde{h}_t$ is the target activation, h_{t-1} is the previous activation and the update gate and reset gates (r_t) are given in Eqs. (8) and (9), respectively [17].

$$z_t = \tanh(Wx_t + \mathrm{U}(r_t \odot h_{t-1}))^j \tag{8}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1})^j \tag{9}$$

A data stream of 10000 packets, each of 64 bits is encoded using an LTE turbo encoder to get the encoded data output of 100000 packets, each of 204 bits, including the tail bits to ensure trellis termination. The RNN model considered for this work is

similar to that of given in [30]. There are 2 layers of bidirectional GRUs. A batch normalization layer follows each GRU layers. The output is a single neuron unit having sigmoid activation. The model is trained for 30 epochs on the training dataset. The performance of GRU based RNN for the decoding of turbo codes is investigated. The modulation considered is Colour Intensity Modulation (CIM) [31] and the communication channel is Additive White Gaussian Noise (AWGN) Channel. The communication system toolbox in MATLAB is used to generate the data in order to train the model and the operation sequence is highlighted in Fig. 4.

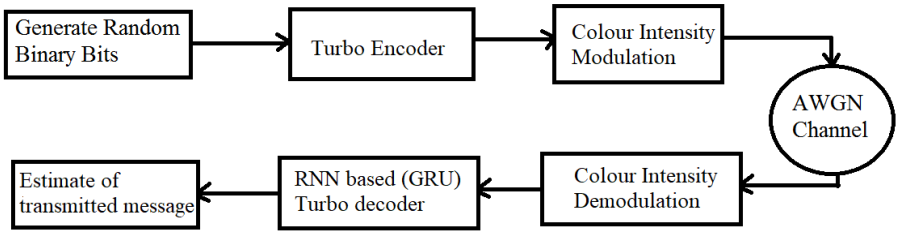

Fig. 4. Simulation of GRU based turbo code decoding.

The BER performance of the proposed system for a range of SNRs between [-2, 2] is shown in Fig. 5. From the simulation results, the performance obtained using RNN performs better than the LTE Turbo decoder for SNRs less than 0.4 db. The major disadvantage of the proposed decoder structure is that at higher SNRs, the BER performance decreases linearly, whereas for conventional decoder, it decreases exponentially. Hence, it can be inferred that by using a hybrid approach of the proposed RNN Turbo decoder at lower SNRs and conventional Turbo decoder at higher SNRs, good performance enhancements can be achieved in modern communication systems.

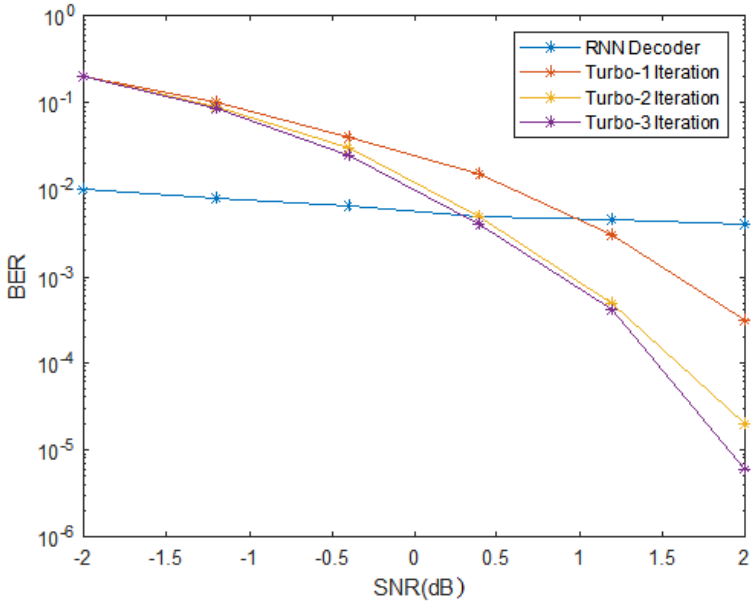

Fig. 5. Decoding performance of the RNN based LTE Turbo decoder.

4. DL design for (16, 8) Polar Code

A feed forward DNN for channel decoding of Polar codes is illustrated in Fig. 6. The performance of the proposed feed forward DNN decoder is compared with the regular SC list decoder. 'K'-bit input information is encoded using standard Polar coder to obtain a 'N'-bit codeword. The codeword is then passed through a CIM modulator [31] over an AWGN channel. The corrupted modulated signal because of noise is passed through a feed forward DNN consisting of L hidden layers. There are N_l number of nodes in the l-th layer having $w_{ij}^{(l)}$ as the weights connecting the feed-forward networks,$i \in [1, N_{l-1}], j \in [1, N_l]$, and nonlinear activation function $\Phi(.)$.

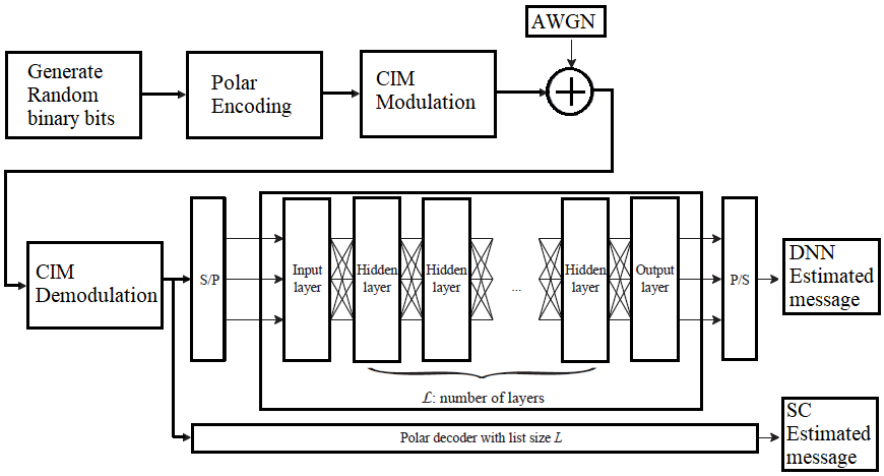

Fig. 6. Polar decoder based on DNN and list decoding.

The performance of the proposed DNN decoder for polar codes by varying the activation functions, size of the neurons in the hidden layer and hidden layer's size is simulated. The decoding performance of the proposed feed forward DNN decoder architecture with Linear, Sigmoid and ReLu activation functions is compared with conventional list decoding as shown in Fig. 7. A (16,8) polar code is chosen for simulation purpose with three hidden layers having nodes 128-64-32.

The trial-and-error method incurs that the increase in the number of hidden layers will improve the decoding BER performance. But this brings increased complexity of training, as the data set used for training and optimizing the connection weights increases. Hence it is required to carefully select the size of hidden layers as to compromise between the training complexity and the BER performance.

The BER performance of the DNN decoder with respect to the number of hidden layers is investigated. Figure 8 shows the BER performance of a two, three, and four hidden layer DNNs, respectively ((64-64), (64-64-64), and (64-64-64-64)), with two list decoding SC algorithms. The results shows that the performance of SC with L=8 is comparable with the four-layer DNN over the entire SNR range. It is found that increasing the number of layers didn't vary much the BER performance. Hence, it is concluded that a four-layer DNN is optimum structure for decoding the polar code. A good performance is also shown by the two layer and three-layer DNNs in low SNR regime.

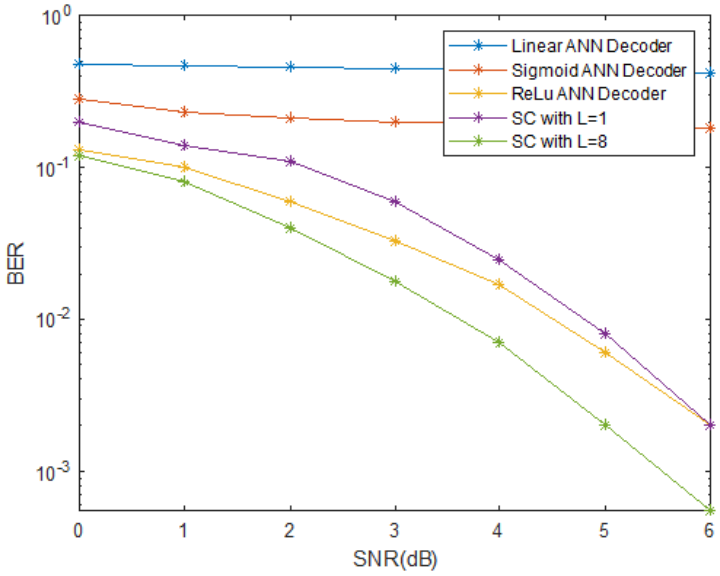

Fig. 7. Comparison of BER performances of polar decoding with different activation functions.

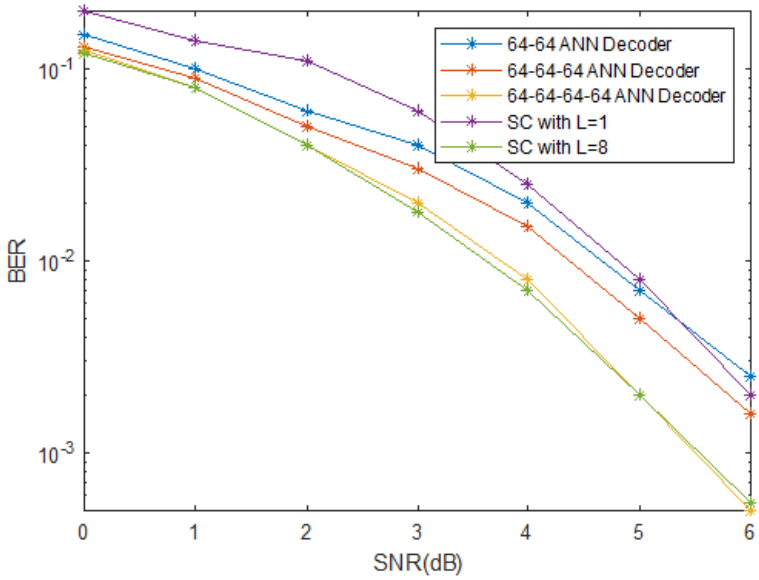

Fig. 8. Effect of different number of hidden layers on BER performance.

The performance of fixed layer DNNs by varying the number of neurons is simulated and the results are analysed. A two-layer DNN with the number of nodes as 16-16, 32-32, 64-64, 128-128 and 256-256 is simulated and the results are shown in Fig. 9. The BER performance shows the SC decoder with list size L=8 yields approximately similar results as that of 128-128 and 256-256 DNN. The 16-16 and 32-32 DNN shows linear BER performance. A good performance is showed by the 64-64 DNN for lower SNR region.

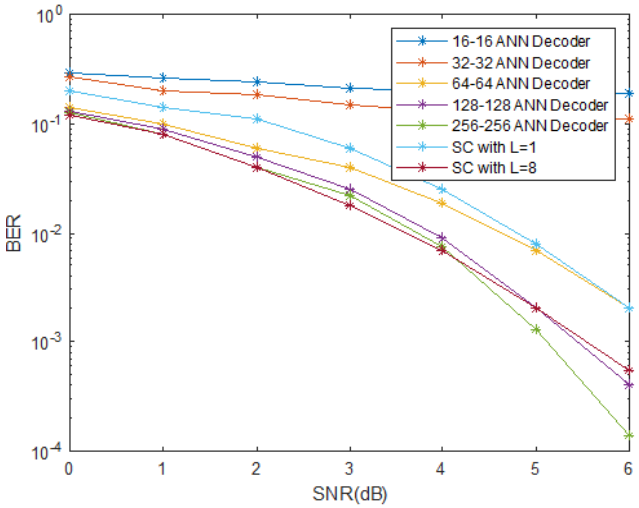

Fig. 9. Effect of different number of nodes on BER performance.

5. Conclusions

This paper gives an approach to decode two well-known FEC codes used in 4G/5G communication technology, namely Turbo Code and Polar Code using DL approach. This paper compares the performance of the proposed RNN based Turbo decoder with the conventional Turbo decoder and DNN based Polar decoder with the SC based Polar decoder. Simulations are carried out in MATLAB environment for training and testing the networks. The results obtained shows the reliability of the proposed architectures for decoding operation. The RNN based on GRUs showed good performance for decoding of Turbo Code at lower SNRs (less than 0.4 dB), whereas at high SNRs the performance reduces linearly as compared to exponential reduction in conventional Turbo decoder. For the Feed-forward DNN of Polar Codes, the architecture when the activation function is ReLU has the best performance. Also, it is seen that, increasing the number of neurons in the hidden layers and the number of hidden layers, improves the performance of the proposed Polar decoder. The Feed-forward DNN showed on par performance when compared to SC based list decoding of Polar Code. The future work of this research could be the application of modern and efficient RNNs such as Neural Turing Machines, Memristive Networks etc. for Turbo decoding operations and implementing the proposed DNN based Polar decoder for higher order Polar codes.

Acknowledgments

The authors acknowledge the support from the Dept. of I&CE, MIT, MAHE, Manipal for providing the necessary facilities and resources to carry out this work in their laboratory.

References

1. Pathak, P.H.; Feng, X.; Hu, P.; and Mohapatra, P. (2015). Visible light communication, Networking, and Sensing: A survey, potential and challenges. *IEEE Communications Surveys & Tutorials*, 17(4), 2047-2077.

2. Gutierrez, J.F.; Hunt, C.E.; and Quintero, J.M. (2014). Visible light communication LED based luminaire for general lighting: state of art. *In Anais do XII Conferência Latino-Americana De Iluminaçao*. Brazil, 163-170.
3. Wang, T.; Wen, C.-K.; Wang, H.; Gao, F.; Jiang, T.; and Jin, S. (2017). Deep learning for wireless physical layer: opportunities and challenges. *China Communications*, 14(11), 92-111.
4. LeCun, Y.; Bengio, Y.; and Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.
5. Gruber, T.; Cammerer, S.; Hoydis, J.; and ten Brink, S. (2017). On deep learning-based channel decoding. *Proceeding of the* 51st *Annual Conference on Information Sciences and Systems*. Baltimore, MD, USA, 1-6.
6. Goodfellow, I.; Bengio, Y.; and Courville, A. (2016). *Deep learning* (Vol. 1). Cambridge: MIT press.
7. Caid, W.R.; and Means, R.W. (1990). Neural network error correcting decoders for block and convolutional codes. *Proceedings of IEEE Global Telecommunications Conference and Exhibition*. San Diego, CA, USA, 1028-1031.
8. Di Stefano, A.; Mirabella, O.; Di Cataldo, G.; and Palumbo. G. (1991). On the use of neural networks for Hamming coding. *Proceeding of IEEE International Symposium on Circuits and Systems*. Singapore, 1601- 1604.
9. Abdelbaki, H.; Gelenbe, E.; and El-Khamy, S.E. (1999). Random neural network decoder for error correcting codes. *Proceedings of International Joint Conference on Neural Networks*. Washington, DC, USA, 3241-3245.
10. Wang, X.-A.; and Wicker, S. (1996). An artificial neural net Viterbi decoder. *IEEE Transactions on Communications*, 44(2), 165-171.
11. Kao, J.W.H.; Berber, S.M.; and Bigdeli, A. (2009). A general rate K/N convolutional decoder based on neural networks with stopping criterion. *Advances in Artificial Intelligence*, Volume 2009 |Article ID 356120, 1-11.
12. Annauth, R.; and Rughooputh, H.C.S. (1999). Neural network decoding of turbo codes. *Proceedings of International Joint Conference on Neural Networks*. Washington, DC, USA, 3336- 3341.
13. Xiao, W.; Al, Y.; and Dong, Z. (2016). The design and implementation of neural network encoding and decoding. *International Journal of Simulation-Systems, Science & Technology*, 17(38), 1-5.
14. Bhavanisankari, S.; Bharathy, G.T.; and Tamilselvi, T. (2019). Error detection in turbo decoding using neural network. *International Journal of Engineering and Advanced Technology*, 9(1S), 218-221.
15. Jiang, Y.; Kim, H.; Asnani, H.; Kannan, S.; Oh, S.; and Vishwanth, P. (2019). Turbo autoencoder: Deep learning based channel codes for point-to-point communication channels. *Proceedings of the* 33rd *Conference on Neural Information Processing Systems*. Vancouver, Canada, 1-11.
16. Devamane, S.B.; and Itagi, R.L. (2022). Recurrent neural network based turbo decoding algorithms for different code rates. *Journal of King Saud University-Computer and Information Sciences*, 34(6), Part A, 2666-2679.
17. Sattiraju, R.; Weinand, A.; and Schotten, H.D. (2018). Performance analysis of deep learning based on recurrent neural networks for channel coding.

Proceedings of the 12*th International Conference on Advanced Networks and Telecommunications Systems*. Indore, India, 1-6.

18. Xu, W.; You, X.; Zhang, C.; and Be'ery, Y. (2018). Polar decoding on sparse graphs with deep learning. *Proceedings of the* 52*nd Asilomar Conference on Signals, Systems, and Computers.* Pacific Grove, CA, USA, 599-603.
19. Xu, W.; Wu, Z.; Ueng, Y.L.; You, X.; and Zhang, C. (2017). Improved polar decoder based in deep learning. *Proceedings of* 2017 *IEEE International Workshop on Signal Processing Systems*. Lorient, France, 1-6.
20. Teng, C.F.; and Wu, A. (2021). Convolutional neural network-aided tree-based bit-flipping framework for polar decoder using imitation learning. *IEEE Transactions on Signal Processing*, 69, 300-313.
21. Teng, C.-F.; and Chen, Y.-L. (2019). Syndrome-enabled unsupervised learning for neural network-based polar decoder and jointly optimised blind equalizer. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 10(2), 177-188.
22. Yuan, C.; Wu, C.; Cheng, D.; and Yang, Y. Deep learning in encoding and decoding of polar codes. *Journal of Physics: Conference Series*, 1060(1), 1-7.
23. TSGR, TS 136 212-V14.2.0-LTE (2017); Evolved universal terrestrial radio access (E-UTRA). Multiplexing and channel coding (3GPP TS 36.212 version 14.2.0 Release 14). *Technical Report*.
24. Tal, I.; and Vardy, A. (2015). List decoding of polar codes. *IEEE Transactions on Information Theory*, 61(5), 2213-2226.
25. Berrou, C.; Glavieux, A.; and Thitimajshima, P. (2003). Near Shannon limit error correcting coding and decoding: Turbo codes. *Proceedings of IEEE International Conference on Communications* (*ICC*). Geneva, Switzerland, 1064-1070.
26. Vaz, A.C.; Nayak, C.G.; and Nayak, D. (2019). Performance comparison between turbo and polar codes. *Proceedings of the* 3*rd International Conference on Electronics, Communication and Aerospace Technology*. Coimbatore, TN, India, 1072-1075.
27. Najim, N.I.; and Ghanim, M.F. (2020). Design and analysis of turbo code for wireless communication networks. *Journal of Engineering Science and Technology* (*JESTEC*), 15(2), 820-830.
28. Arikan, E. (2009). Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7), 3051-3073.
29. Balatsoukas-Stimming, A.; Parizi, M.B.; and Burg, A. (2015). LLR-based successive cancellation list decoding of polar codes. *IEEE Transactions on Signal Processing*, 63(19), 5165-5179.
30. Kim, H.; Jiang, Y.; Rana, R.; Kannan, S.; Oh, S.; and Viswanath, P. (2018). Communication algorithms via deep learning. *Proceedings of the* 6*th International Conference on Learning Representations*. Vancouver, Canada, 1-17.
31. Vaz, A.C.; Nayak, C.G.; and Nayak, D. (2020). Colour light intensity-based modulation scheme for visible light communication employing turbo codes. *Journal of Advanced Research in Dynamical and Control Systems*, 12(3 Special Issue), 1051-1060.