# An iterative BP-CNN Architecture for Channel Decoding (Extended Abstract)

Fei Liang
SeniorMember,IEEE
Cong Shen
SeniorMember,IEEE
Feng Wu
Fellow,IEEE

Laboratory of Future Networks, School of Information Science and Technology, University of Science and Technology of China, Hefei, China

## 1 Introduction

In this paper, authors have discussed the advancements in the methods of noise reduction at the receiver side. They have suggested update in channel decoding method with deep learning technique. The main objective of this paper is to find a new loss function for better accuracy noise estimation and better results for normality test.

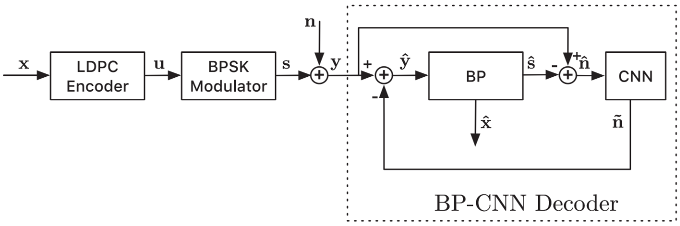


Figure 1: Block Diagram

## 2 System

The complete system can be divided into 4 parts, the first one is the pre-processing of the signal, followed by modulation and transmission of the signal these constitutes transmitter section. The third part is of receiving the signal, demodulating it to intermediate frequency and last one is decoding the signal and making it noise free, these constitutes the receiver section. Fig.1. shows the overview of the system.

The system begins with the pre-processing of the signal, which includes analog to digital conversion and amplification of the signal. At the end of this step we have signal 'x' of length 'K'. 'x' is then encoded to a binary code word 'u' of length 'N' through a Low Density Parity Check(LDPC) encoder. In next step, this code word 'u' is mapped to a symbol vector 's' through the BPSK(Binary Phase Shift Keying) modulation. These BPSK symbols are passed through the channel with the additive Gaussian noise. The channel noise vector, denoted as 'n' of length 'N' is modeled as the Gaussian random vector with the auto-correlation matrix 'sigma'. Now comes the receiver part. The received signal is the modulated signal with some noise added with it. It is believed that noise is generated from the same sources, thus the noise vector will follow a Gaussian distribution and will have some correlation. The received signal is firstly demodulated using BPSK demodulator to bring it to message signal of original frequency. This message signal now is given to the decoder. This paper has discussed the decoder named Belief Propagation decoder(BP) along with neural network block which a CNN. This cumulatively they have termed as BP-CNN.

Belief propagation, also known as sum-product message passing, is a message-passing algorithm for performing inference on graphical models, such as Bayesian networks and Markov random fields. It calculates the marginal distribution for each unobserved node (or variable), conditional on any observed nodes (or variables). It mainly passes messages between neighboring nodes by joining the messages from all the neighbors except the receiving node. Where as in deep learning, a convolutional neural network (CNN, or ConvNet) is a class of deep neural networks, most commonly applied to analyzing visual imagery. CNN is used because of its strong capability to extract local features. The correlation in noise(noise is assumed to follow Gaussian distribution) is used as a feature for CNN which happens because of filtering, oversampling and device noise. This noise correlation vector is 1-D vector of size Nx1. Using this vector, feature mapping is done for further layers. The sequence of this CNN network structure is 4;9,3,3,15;64,32,16,1. Xaviert Initialisation is used for assigning the weights, this method helps to keep the variance same i.e. it keeps the signal exploding to high value or vanishing it to zero. The training data is generated under multiple Signal to Noise Ratio ratio(SNR) eg.0, 0.5, 1,1.5, 2, 2.5, 3 etc, higher the value lesser is the noise component. To train the network Mini-batch gradient descent method is used. Each mini-batch contains 1400 blocks of data and data of SNR occupies the same proportion in one mini-batch. For optimisation, ADAM optimisation technique is used because of its high performance gains in terms of speed of training. The performance of this system depends upon the number of iteration and type of noise getting added as it is expected for the noise to have correlation and follows a Gaussian distribution, on which CNN can predict the noise accurately which can be cancelled in further steps. In the proposed iterative BP-CNN method, the input is a 1-D vector. The 'k1' feature map is generated from the input data along with ReLU(Rectified Linear Unit) as the activation function in order to introduce non-linearity. CNN is trained to provide residual noise which is beneficial for the BP(Belief Propagation) decoding. Paper has discussed two strategies for training the CNN and concatenating it with BP. The authors have discussed 2 methods for training CNN, first, Baseline BP-CNN, a standard method which can be implemented using a quadratic loss function in CNN training and the other one is Enhanced BP-CNN which uses strategy to depress the residual noise power and simultaneously let the residual noise follow a Gaussian distribution. The loss function(Jarque-Berg Test)[1] which is used to train the CNN is as follows.

$$Loss_B = \|r\|^2/N + \lambda(S^2) + 1/4(C-3)^2$$

$$S = 1/N\sum_{i=1}^{N}(r_i-\bar{r})^3/(1/N\sum_{i=1}^{N}(r-\bar{r})^2)^{3/2}$$

$$C = 1/N\sum_{i=1}^{N}(r_i-\bar{r})^4/(1/N\sum_{i=1}^{N}(r-\bar{r})^2)^2$$

Here, S and C are skewness and kurrosis respectively, $\lambda$is scaling factor, r is residual noise given as r=n-n

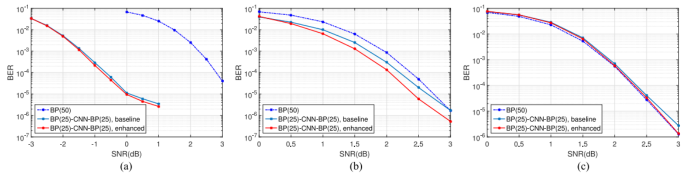


Figure 2: Bit Error Rate(BER) vs. Signal to Noise Ration(SNR)

## 3 Conclusion

Author's proposed method concatenates BP with CNN and iterates between them. The results shown in paper of performance gain w.r.t. different values of correlation measure and scaling factor of normality test, which increases with the iterations. The results showed the effectiveness of proposed iterative BP-CNN decoder. Fig.2.shows the performance comparison of BP-CNN with the standard BP decoding. Only one iteration between CNN and BP decoder is executed. The numbers in the brackets denote the BP iterations. The SNRs corresponding to the Shannon capacities for =0.8,0.5, and 0 are 6.9 dB, 2.4 dB and 0.4 dB, respectively. (a) =0.8 , strong correlation. =0.1. (b) =0.5, moderate correlation. =10. (c) =0 , no correlation. =10.

[1] T. Thadewald and H. Buning. Jarque–bera test and its competitors for testing normality: A power comparison. J. Appl. Statist., vol. 34, no. 1, pp. 87–105.