



Frame Detection with Deep Learning

Mete Yildirim^{1*}, Radosveta Ivanova Sokullu¹

¹ Department of Electrical and Electronics Engineering, Ege University, 35030, Bornova, İzmir, Turkey

*mete.yildirim@mail.ege.edu.tr

*Orcid: 0000-0001-6335-4752

Received: 25 February 2020

Accepted: 21 May 2021

DOI: 10.18466/cbayarfbe.693942

Abstract

Deep learning has become a way of solution for the realization of complex computations. As electronic communication starts to use more complex channels, the systems need to handle tough computations. For this reason, research on the use of deep learning in communication has increased recently. These researchers aim to realize many applications used in communication with deep learning. Frame detection is one of the first things a receiver must handle, and it may require a lot of hard computations. Deep learning-based frame detection can be an alternative approach. This study aims to build models that perform frame detection with deep learning. The proposed models provide the performance of correlation-based frame receivers commonly used for frame detection. The mean square root error of the prediction deviation is used as an evaluation metric to compare the proposed model to classic systems.

Keywords: Communication, correlator, deep learning, frame detection, neural network

1. Introduction

Many modern communication systems transmit information in the form of a packet or frame in order to share the transmission medium. Frame structure usually contains information that determines its start and endpoint. The receiver must detect these points correctly. This process is called frame detection [1–2].

Frame detection is vital to the performance of the system. The error to be made in frame detection can cause a lot of other synchronization problems in the system such as symbol, frequency, and etc. which in turn reduces the performance of the system.

Mostly, the good frame detection depends on the algorithm used and the structure of the frame. A well-designed frame should not increase the overhead information while facilitating frame detection, nor should it be in a format that costs more processing time and energy.

The use of neural networks, which is the most primitive version of the deep learning model, is not new. [3] and the references within summarize these studies. These studies cover many important topics for communication systems such as modulation, demodulation, detection, synchronization, coding. These studies have not produced very successful results. However, the deep learning models obtained with multi-layered neural

networks succeeded in producing successful results in communication systems. [4–6] are some of the review papers examine many excellent applications of deep learning to electronic communication. [7–14] show very successful machine learning-based communication. [9] combines all communication blocks in one entity and put forward a different approach for digital communication. [7] implements deep learning-based communication with software defined radio on AWGN channels. Generally, these papers try to optimize overall performance of the digital systems and do not let us know if deep learning can handle some very important receiver tasks such as frame detection individually and how good it is in these tasks. This motivates us to examine a deep learning-based frame detection. In this study, we will consider received signals that contains a frame in various noise levels. The proposed models detect frame beginning for various size of preamble information in various level of signal to noise ratio and the performance of the models compared to the correlation-based detector.

2. Deep Learning Foundation

Deep learning is performed with deep neural network (DNN) that is very similar the shallow neural network (NN) in structure. Unlike NN, DNN generally have many layers. The connection between these layers may differ from NN and can be very complicated. DNN has

different approaches than NN that help to create many successful applications to the various field. The calculation in NN is done by a neuron. Being the most basic unit of NN, a neuron does a simple calculation on the given input given as $z = w^T x + b$. The output is estimated with an activation function $a = g(z)$ [15-17]. The activation function, $g(\cdot)$, is generally non-linear. Some of them are named as “relu, tanh, sigmoid” etc. These non-linear function gives power to the model for non-linear output estimation. The structure that combines many neurons named layer. The neurons in a layer do not have inner connection and works independently.

The machine learning generally subcategorized in two. They are supervised and unsupervised learning. In this study we only consider supervised learning. The supervised learning is performed based on known input and output. The model sets the inner parameters by itself to get desired outputs for the given inputs. This is done on many samples that is known as training samples. The best inner parameters are saved and used for validation data that is used to understand the model training mood. If the model is not undertrained nor over trained, it becomes ready to use for prediction on new datasets. For satisfactory result, the network must be provided enough training examples $(x(i), y(i))$. The samples can be applied the model on vector based known feature vector. The computation of a neuron for the input features vector x is given as $z = w^T x + b$, where w is weighting vector and b is constant, hence a layer output given as is $Z = W^T x + b$, where W is a matrix, b is a constant vector. After calculation nonlinear output of neurons with the one of mentioned activation functions, the output is applied to next layer. In this calculation, W and b are randomly initialized to small numbers. That calculation goes on until the last layer. The last layer of model must have neurons as many as the number of classes in classification problem. In other words, each class is represented with a neuron on last layer where the calculated values are converted to probability by using softmax function etc. In general, a NN is mapping of N_i dimension of input to N_o dimension of output given as $f(x; W, b): R^{N_i} \rightarrow R^{N_o}$. The difference between real output and model output is calculated with a cost function.

$$\sum_{i=1}^m (y^{(i)} \log a^{[L](i)} + (1 - y^{(i)}) \log (1 - a^{[L](i)})) \quad (2.1)$$

The cost function is chosen according to the need. (2.1) shows a cost function for binary classification named as logistic regression log likelihood [18].

The cost function can be maximized using gradient decent that is called the training of network. The more on deep learning, such as multinomial classification, softmax and etc. can be found in [19].

3. Frame Detection Model

A frame is generally a joint structure of the preamble and an information message. A preamble known as marker is mostly a predesigned sequence of bits while the information message is a random sequence of symbols from the used alphabet. Sometimes the preamble itself can be made from the message to be transmitted for reducing waste of resources as given in [20]. Frame detection can be examined in two basic categories without loss of generality. These are correlation-based receiver (CBR) and maximum likelihood-based receiver (MLR)[2]. In the CBR, the received sequence is correlated with the local preamble and the correlation peak is used to determine starting position of the frame. The weaknesses of CBR can be considered as long processing time, saving a copy of preamble locally and sensitivity to frequency variation. The MLR shows better performance in the case of frequency deviation but considered as a costly way of handling frame detection.

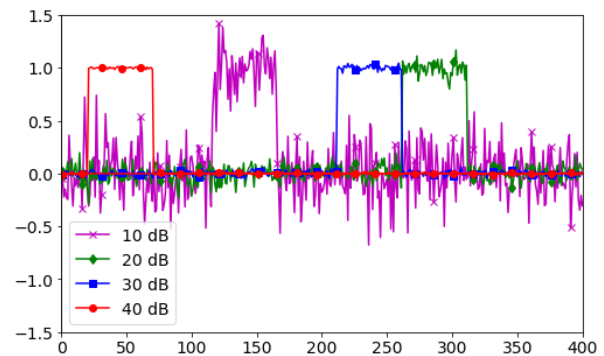


Figure 1. The samples of received signals.

In this study, we consider the frame detection with deep learning and examine the different models and compare their performance to the basic correlator-based detector. The two are compared in terms of the deviation from the actual frame starting position by evaluating the deviation as a mean squared estimation error. The received signal is corrupted by the noise that reduces signal-to-noise ratio. The Fig.1 shows four signals received by a receiver at different time with various signal to noise ratio. This information could be considered a preamble to actual message signal to be transmitted. Since the message is a sequence of random symbols, it is preferred not to show in the figure for the sake of better visualization clarity. The receiver has to find where the frame starts. We consider the correlation-based frame detector against the deep learning-based frame detector. While the deep learning-based receiver does not need to know what kind of preamble is sent, the correlation-based receiver needs to know the exact preamble. Therefore, the preamble between the receiver and the transmitter must be agreed beforehand which can be considered handicap of correlated-based receiver. Further, the CBR may require long processing

time. For the deep learning, it only requires a good training before it is used.

The CBR correlates the received information with locally saved one and used the peak point as the starting position of the frame. The proposed DL model uses a softmax layer at the output and calculates the probability of every possible position for the preamble in the received corrupted sequence and choses the highest probability as the starting position of the frame. The input to model is the received sequence. As it could be possible to consider real and imaginary part separately, only the real part is considered in this study which turns out to provide satisfactory results.

The proposed deep learning-based receiver uses fully connected layers. It has 6 hidden layers, dropout=0.2, number of neurons per layer 512, training size between 100000-150000. While it is possible to provide deeper and larger NN, to have moderate training time, parameters are used. First, we provide the figures that shows the model neither overfits nor underfits. The overfitting and underfitting are two important issues in the training process of a NN model. The loss and the accuracy for the validation and the training datasets are considered to decide a well-fitting model. In Fig.1 accuracy and loss graphs are given for various signal to noise ratio against number of the epochs. It is clearly seen that different signal to noise ratio requires different number of epochs. For example, 5 epochs are enough for -10dB while it becomes 30 epochs for 5dB. Generally, it requires more epochs around 0-10dB.

The expected outcome for the model to generalize well is that the accuracy should be on the rise while for the loss is getting smaller and the gap between the validation and the training set should be moderately

close for both as the number of the epochs increase. It is understood from the graphs on Fig.2 that around 15 epochs are enough for well generalized model. The proposed model is trained for various SNR and preamble size. The Fig.3 shows DL model prediction versus CBR prediction error in terms of mean squared deviation. The two compared for the various preamble size in the range of 5 to 40 symbol for the signal to noise ration from -15dB to 10dB. It is seen from the figure; DL model can put so close performance to CBR with even a simple model given above. Both receivers give zero deviation error above 10dB.

Deep learning model can be in various forms. Some forms show better performance than others. The wisdom behind it cannot be explained mathematically. Hence, developing a good deep learning model is achieved through trial and error. The number of neurons used in model is considered part of parameters used in a model. Apart from the total number of neurons, their distribution can be matter. In the study the distribution of neurons to the hidden layers are considered as one of the parameters to adjust. Hence, we try to find out the best distribution to the hidden layers. Several distribution approaches are tested with various size of preamble. After many trainings of these models, we have seen that even distribution is the best choice in term of classification accuracy. The even distribution means that all the hidden layers have same number of neurons. The output of the proposed model is shown in Fig.4 that is tested for two different preamble sizes in the range of -15dB to 15dB signal to noise ratio. The Fig.4 shows the performance of the model with even and uneven distribution of neurons to the hidden layers for preamble size 5 and 10. The model with the even distribution generally shows better performance

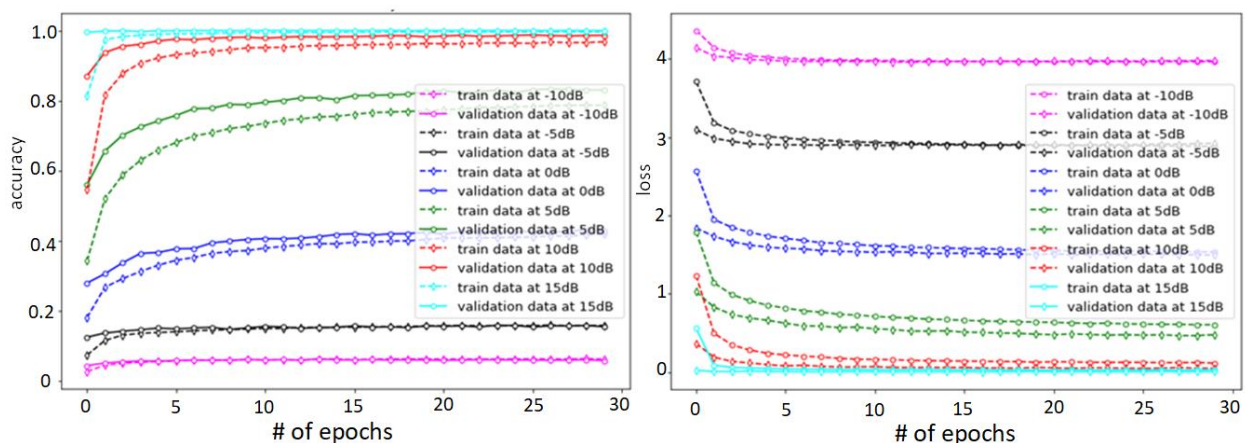


Figure 2. The loss and the accuracy of deep learning-based receiver.

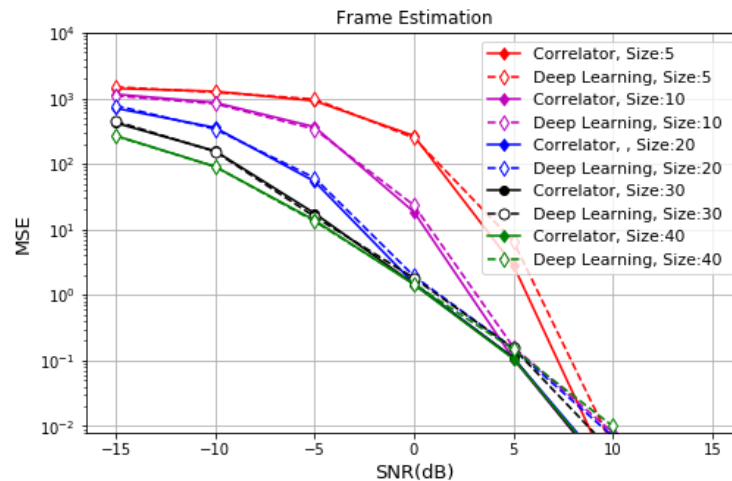


Figure 3. The Deep Learning-based and the correlator-based detector performance.

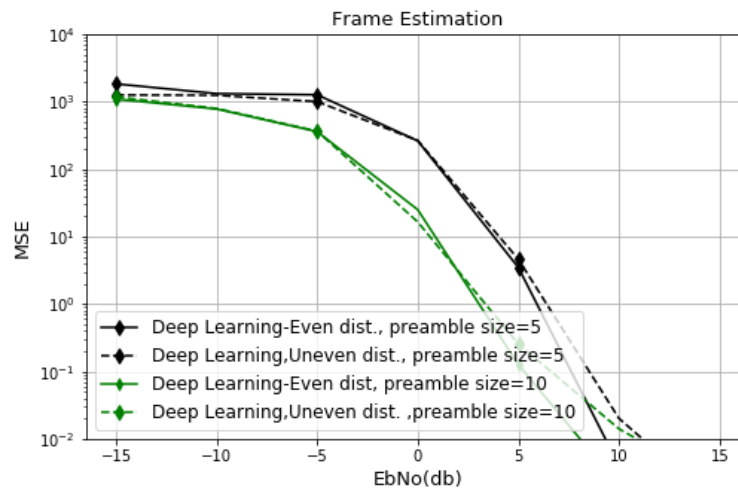


Figure 4. Distribution of neurons

4. Conclusion

A deep learning model is designed to make frame detection. This model has been shown to be well-fitting. For this purpose, loss and accuracy graphs in validation and training datasets are given by simulation. Then, frame detection performances of these models were compared with correlation-based system. The mean square error of deviation is used as comparison metric. Then, some parameters of the model are adjusted and the best performing model for frame detection is formed. The model is observed to be successful in comparison to correlator-based detector. The deep learning-based detector is considered to superior to correlator-based detector since the second must have the preamble beforehand for detection while the first does not need to know it. This is considered as one of the proposed model superiorities to the correlator-based detector. The second superiority is that the correlator-

based detector has to do heavy correlation calculations for every detection while the deep learning-based detector needs only offline training. This helps to reduce processing time and memory requirements seriously, requires less computing power and less power consumption at the receiver. These advantages come from the fact that DL based model just need to save weighted parameters that is used in some simple mathematical calculation to estimate the frame position. Observing the contribution of deep learning-based model frame detection, the conclusion we arrive is that the future communication systems must consider to use the DL based technologies.

Acknowledgement

This study was carried out in the Department of Electrical and Electronics Engineering of Ege University within the scope of the project supported by TUBITAK 2218.



Author's Contributions

Mete Yıldırım: Conceptualization, Methodology, Validation, Simulation, Formal analysis, Investigation, Writing - original draft, Writing - review, and editing.
Radosveta Ivanova Sokullu: Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing - review & editing, Supervision.

Ethics

There are no ethical issues after the publication of this manuscript.

References

1. Massey, J. 1972. Optimum frame synchronization. *IEEE transactions on communications*; 20(2): 115–119.
2. Scholtz, R. 1980. Frame synchronization techniques. *IEEE transactions on communications*; 28(8): 1204–1213.
3. Ibnkahla, M. 2000. Applications of neural networks to digital communications—a survey. *Signal processing*; 80(7): 1185–1215.
4. Qin Z, Ye H, Li G, Juang F. 1972. Deep learning in physical layer communications. *ArXiv*: 1807.11713
5. Simeone, O. 2018. A very brief introduction to machine learning with applications to communication systems. *IEEE Transactions on Cognitive Communications and Networking*; 4(4): 648–664.
6. Wang T, Wen CK, Wang H, Gao F, Jiang T. 2017. Deep learning for wireless physical layer: Opportunities and challenges. *China Communications*; 14(11): 92–111.
7. Dörner S, Cammerer S, Hoydis J, Brink S. 2017. Deep learning based communication over the air. *IEEE Journal of Selected Topics in Signal Processing*; 12(1): 132–143.
8. O'Shea T, Hoydis J. 2017. An introduction to deep learning for the physical layer. *IEEE Transactions on Cognitive Communications and Networking*; 3(4): 563–575.
9. O'Shea T J, Karra K, Clancy T C. 2016. Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention. *arXiv*: 1608.06409.
10. TO'Shea T J, Roy T, West N, Hilburn B C. 2018. Physical Layer Communications System Design Over-the-Air Using Adversarial Networks. *ArXiv*: 1803.03145.
11. O'Shea T J, Hoydis J. 2017. An introduction to machine learning communications systems. *ArXiv*: 1702.
12. O'Shea T J, Erpek T, Clancy T.C. 2017. Deep learning based MIMO communications. *arXiv*:1707.07980.
13. TO'Shea T J, Roy T, West N, Hilburn B C. 2018. Physical Layer Communications System Design Over-the-Air Using Adversarial Networks. *ArXiv*: 1803.03145.
14. O'Shea T J, Hoydis J. 2017. An introduction to machine learning communications systems. *ArXiv*: 1702.
15. Ramachandran, P., Zoph, B. and Le, Q.V., 2017. Searching for activation functions. *arXiv preprint arXiv:1710.05941*.
16. Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. 2018. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*.
17. Agostinelli, F., Hoffman, M., Sadowski, P., & Baldi, P. 2014. Learning activation functions to improve deep neural networks. *arXiv preprint arXiv:1412.6830*.
18. Nie, F., Hu, Z., & Li, X. 2018. An investigation for loss functions widely used in machine learning. *Communications in Information and Systems*, 18(1), 37-52.
19. LeCun Y, Bengio Y, Hinton G. 2015. Deep learning. *Nature*; 521(7553): 4362
20. Schmidl T M, Cox D C. 1997. Robust frequency and timing synchronization for OFDM. *IEEE transactions on communications*; 45(12): 1613–1621.