Deep Learning based Channel Estimation and Equalization Schemes for FBMC-OQAM Systems

*Note: This paper is written as a part of the course EE 678 - Wavelets in Spring 2021 taken by Prof. Gadre at Indian Institute of Technology Bombay

Amol Girish Shah
Department of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
18D070005@iitb.ac.in

Mantri Krishna Sri Ipsit
Department of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
180070032@iitb.ac.in

***Abstract*—Data driven approaches are finding their ways in many areas in the recent times. Any wireless communication system involves huge amount of data. Deep learning based approaches have given promising results in other domains like computer vision and natural language processing. In this paper, we would like to reflect on how such methods can be adapted to the channel estimation and equalization problem in the Filter Bank Multi Carrier - Offset Quadrature Amplitude Modulation (FBMC-OQAM) systems. We provide details of three such methods.**

***Index Terms*—FBMC, channel estimation, recurrent neural networks, long short-term memory, convolutional neural networks, residual neural networks, deep neural networks, deep learning, function approximation**

I. Introduction

FBMC is considered as the modulation technology of the next generation wireless communication systems. FBMC has higher spectral efficiency, higher flexibility and low out-of-band energy leakage than orthogonal frequency division multiplexing (OFDM). But the main drawback in FBMC is the increased complexity of the equalizers due to the intrinsic interference.

A variety of channel estimation schemes have been proposed specifically to tackle the intrinsic interference in FBMC. These schemes can be broadly classified into two categories: 1) scattered pilot-based schemes [1] 2) preamble-based schemes [2], [3] . The Interference Approximation Method is very popular among the preamble-based schemes and is widely studied.

Recently, deep learning methods are beating human-level performance in the fields of computer vision, natural language processing, robotic intelligence, etc,. Deep learning applications are becoming more common in the field of wireless communications [4]. Promising results have been obtained for the problems like channel estimation [5], [8], restoring channel state information (CSI) in the massive MIMO systems [6], reduction of high peak-to-average power ratio (PAPR) of OFDM signals [7], etc.,

In this paper, we would like to discuss and reflect upon three deep learning based channel estimation and equalization schemes for FBMC-OQAM systems, namely

1) Residual Network - Deep Neural Network (ResNet-DNN) based scheme [9]
2) Convolutional Neural Network - Recurrent Neural Network (CRNN) based scheme [10]
3) Deep Bidirectional Long Short-Term Memory (BLSTM) recurrent neural network based scheme [11]

The rest of the paper is organized as follows: In section II we provide the necessary background information and then in section III we provide a summary for each of the three methods. In section IV we propose potential extensions and improvements. In section V we provide our conclusion.

II. Background information

In this section, the background theory is provided which is required to understand the paper.

A. FBMC-OQAM System Model

The discrete-time FBMC-OQAM transmitted signal with N subcarriers is given by

$$s[l] = \sum_{m=-\infty}^{\infty} \sum_{n=0}^{N-1} x_{m,n} e^{j\theta_{m,n}} e^{j2\pi nl/N} g\left[l - m\frac{N}{2}\right] \quad (1)$$

where $x_{m,n}$ is the real-valued transmitted data at n-th subcarrier (frequency) of the m-th symbol (time). Two such symbols are combined to form complex symbols and this symbol is modulated by the 2^{2r}-ary quadrature amplitude modulation (QAM). $\theta_{m,n} = \frac{\pi}{2}(m+n)$ and the transmitter basis pulse $g_{m,n}[l]$ is a time and frequency shifted symmetrical and real-valued propotype filter $g[l]$. If O is the overlapping factor, the length of $g[l]$ is ON.

Perfect reconstruction of the real-valued frequency-domain data $\hat{x}_{m_0,n_0}$ at the TF point (m_0, n_0) under ideal channel conditions is given by

$$\hat{x}_{m_0,n_0} = \sum_{m=-\infty}^{\infty} \sum_{n=0}^{N-1} x_{m,n} \langle g_{m,n}[l], g_{m_0,n_0}[l] \rangle_{\mathcal{R}} \quad (2)$$

The prototype filter is designed to satisfy the real-domain orthogonal condition. This is in accordance with the Balian-Low theorem. Any inter-symbol or inter-carrier interference

manifests itself as a purely imaginary number. For simplicity, we denote the inner product of $g_{m,n}[l]$ and $g_{m_0,n_0}[l]$ by

$$\xi_{m,n}^{m_0,n_0} = \langle g_{m,n}[l], g_{m_0,n_0}[l] \rangle = \sum_{l=-\infty}^{\infty} g_{m,n}[l]\, g*_{m_0,n_0}[l] \quad (3)$$

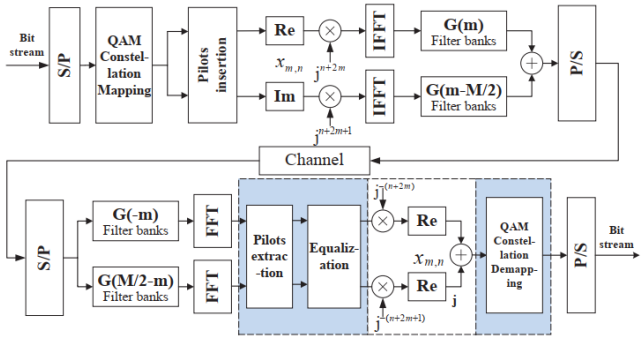

Fig. 1. The schematic of the FBMC-OQAM scheme implemented by IFFT/FFT [22]

B. The Problem of Channel Estimation in FBMC

A symbol at the TF point (m, n) is subject to interferences from the symbols at its surrounding points in the TF plane. This is not a problem when we are estimating a real-valued channel since the interference is in purely imaginary domain. However, for the estimation of a complex valued channel (which is the case most of the times), the channel coefficient estimated by the pilots is the superposition of the actual channel coefficient and the interferences from the surrounding symbols. The interferences cannot be eliminated if the values of the surrounding symbols are random, which will reduce the accuracy of channel estimation.

1) Interference Approximation Method: If the channel is stable in both frequency and time domains [2], [21], then $\hat{x}_{m_0,n_0}$ can be written as

$$\hat{x}_{m_0,n_0} \approx H_{m_0,n_0} \underbrace{(x_{m_0,n_0} + jx_{m_0,n_0}^{(i)})}_{x_{m_0,n_0}^{(c)}} + \eta_{m_0,n_0} \quad (4)$$

where H_{m_0,n_0} denotes the channel coefficient at (m_0, n_0), η_{m_0,n_0} is the noise component, and $jx_{m_0,n_0}^{(i)}$ represents the corresponding imaginary interferences,

$$x_{m_0,n_0}^{(i)} = \sum_{(m,n)\in\Omega_{m_0,n_0}} x_{m,n}\, \xi_{m,n}^{m_0,n_0} \quad (5)$$

where Ω_{m_0,n_0} denotes the neighbourhood points of (m_0, n_0) in the TF plane. The datas at (m_0, n_0) and Ω_{m_0,n_0} are usually fixed as preamble or block pilots so that the value of $x_{m_0,n_0}^{(c)}$ can be approximated and can be treated as a pseudo pilot to estimate the channel coefficient $\hat{H}_{m_0,n_0}$ at (m_0, n_0)

$$\hat{H}_{m_0,n_0} \approx H_{m_0,n_0} + \frac{\eta_{m_0,n_0}}{x_{m_0,n_0}^{(c)}} \quad (6)$$

Usually the prototype filter is well designed and well localized in the TF plane so that the majority of interference occurs due to the close meighbours of (m_0, n_0) i.e., $\Omega_{m_0,n_0} = \{(m_0 \pm 1, n_0 \pm 1), (m_0 \pm 1, n_0), (m_0, n_0 \pm 1)\}$

C. Residual Networks

Deep Neural Networks are becoming deeper and more complex. It has been proved that adding more layers to a Neural Network can make it more robust for image-related tasks.The tendency to add so many layers by deep learning practitioners is to extract important features from complex images. So, the first layers may detect edges, and the subsequent layers at the end may detect recognizable shapes, like tires of a car. But if we add more than 30 layers to the network, then its performance suffers and it attains a low accuracy. This is contrary to the thinking that the addition of layers will make a neural network better. This is not due to overfitting, because in that case, one may use dropout and regularization techniques to solve the issue altogether. It's mainly present because of the popular vanishing gradient problem [12] (the vanishing gradient problem is encountered when training artificial neural networks with gradient-based learning methods and backpropagation. In such methods, each of the neural network's weights receives an update proportional to the partial derivative of the error function with respect to the current weight in each iteration of training. The problem is that in some cases, the gradient will be vanishingly small, effectively preventing the weight from changing its value. In the worst case, this may completely stop the neural network from further training. [13]) In Residual Neural networks or Res-Nets, as they are popularly called, are neural networks which make use of skip connections or shortcuts which jump over some layers. The problem of vanishing gradients is taken care of when we use the activations from previous layers which helps in speeding up learning during the initial phase.

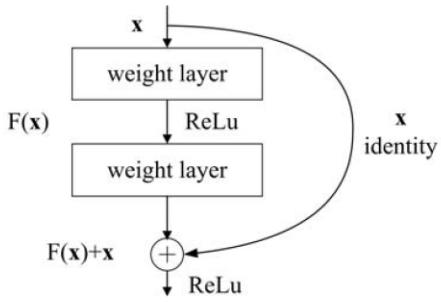

Fig. 2. ResNet standard unit [9]

D. Convolutional Recurrent Neural Network(CRNN)

Convolutional Recurrent Neural Networks are a combination of two of the most prominent neural networks which involves CNN(convolutional neural network) followed by the RNN(Recurrent neural networks). While Convolutional Neural Networks help us at extracting relevant features in the image,

Recurrent Neural Networks help the NNet to take into consideration information from the past in order to make predictions or analyze. The Convolutional Neural Network analyzes the image, sending it to the recurrent part of the important features detected. The recurrent part analyzes these features in order, taking into consideration previous information in order to realize what are some important links between these features that influence the output [15].

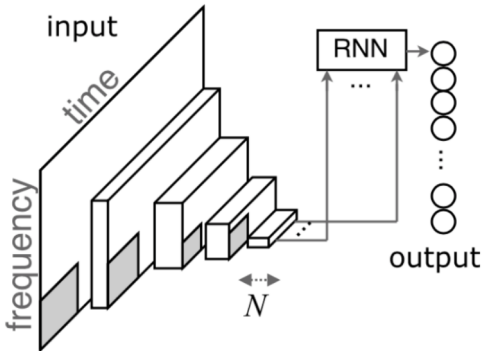

Fig. 3. CRNN basic model [16]

In radio communications systems, many recovery processes can be thought of in terms of invariance to linear mixing, rotation, time shifting, scaling, and convolution through random filters (with well characterized probabilistic envelopes and coherence times). This is analogous to similar learning invariance which is significantly addressed in vision domain learning where matched filters for specific items or features in the image may undergo scaling, shifting, rotation, occlusion, lighting variation, and other forms of noise. Thus we leverage the shift-invariant properties of the convolutional neural network to learn matched filters reducing temporal variations that recover the transmitted signals. We feed the learned filters into a subsequent layer for temporal modeling by utilizing LSTM cells, which specify the RNN part. The layer which consists of LSTM cells is called LSTM layer. Basically, the LSTM layer learns temporal dependency by memorizing the previous internal state and adding it to the current state at every single time step, which is what recurrent means. [10]

E. Deep Bidirectional LSTM RNN

1) RNN: In a classical neural network, there are only weighted connections between the layers and the neurons of each layer are considered to be independent. Whereas in a RNN, the neighbouring neurons are also connected with weights. This type of architecture is useful when dealing with sequential or time series data. For an input sequence $\boldsymbol{x} = (\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_T)$ and the output target sequence $\boldsymbol{y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_T)$, a RNN computes the hidden vector sequence $\boldsymbol{h} = (\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_T)$ using the following equations at each time step:

$$\boldsymbol{h}_t = g(\boldsymbol{W}_{xh}\boldsymbol{x}_t + \boldsymbol{W}_{hh}\boldsymbol{h}_{t-1} + \boldsymbol{b}_h) \tag{7}$$

$$\boldsymbol{y}_t = s(\boldsymbol{W}_{hy}\boldsymbol{h}_t + \boldsymbol{b}_y) \tag{8}$$

Here $\boldsymbol{W}_{xh}$ is the weight vector relating between $\boldsymbol{x}$ and $\boldsymbol{h}$.

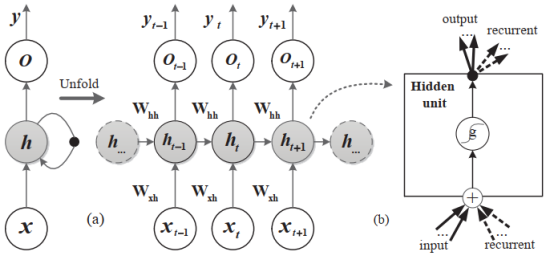

Fig. 4. (a) General architecture of a standard RNN with feedback connections and unfolded in time for three time steps. (b) A hidden unit of RNN. [11]

Other weight vectors are defined analogously. g is the Sigmoid function and s is the output layer function, it can also be a neural network. The structure of a classical RNN is shown in figure 4. It is evident from the equations at each time step that a RNN captures and makes use of the information and dependencies present in all the information that has passed through it. But it is often the case in many scenarios like text translation and text completion, the context of a word depends on both the words before and after it in the sentence, i.e., the present information depends on both the past and the future values. To capture this effect, bidirectional RNNs (BRNNs) have been proposed.

2) BRNN: As the name suggests, a BRNN processes the data in both forward and backward directions with two seperate hidden layers which are fed forward to the same output layer. At each time step, the transformations occuring in a BRNN are as follows:

$$\overrightarrow{\boldsymbol{h}}_t = g(\boldsymbol{W}_{x\overrightarrow{h}}\boldsymbol{x}_t + \boldsymbol{W}_{\overrightarrow{h}\overrightarrow{h}}\overrightarrow{\boldsymbol{h}}_{t-1} + \boldsymbol{b}_{\overrightarrow{h}}) \tag{9}$$

$$\overleftarrow{\boldsymbol{h}}_t = g(\boldsymbol{W}_{x\overleftarrow{h}}\boldsymbol{x}_t + \boldsymbol{W}_{\overleftarrow{h}\overleftarrow{h}}\overleftarrow{\boldsymbol{h}}_{t-1} + \boldsymbol{b}_{\overleftarrow{h}}) \tag{10}$$

$$\boldsymbol{y}_t = s(\boldsymbol{W}_{\overrightarrow{h}y}\overrightarrow{\boldsymbol{h}}_t + \boldsymbol{W}_{\overleftarrow{h}y}\overleftarrow{\boldsymbol{h}}_t + \boldsymbol{b}_y) \tag{11}$$

where $\overrightarrow{\boldsymbol{h}}_t$ denotes the forward hidden sequence and $\overleftarrow{\boldsymbol{h}}_t$ denotes the backward hidden sequence. The structure of a classical BRNN is shown in figure 5

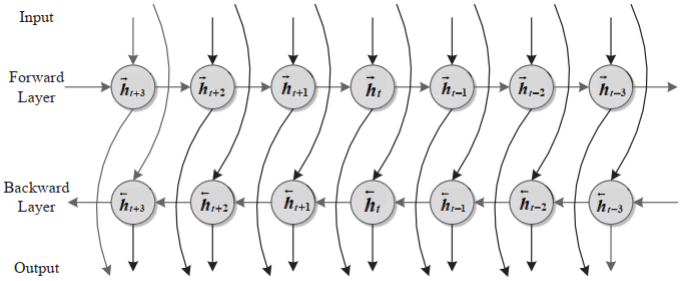

Fig. 5. The structure of bidirectional RNN (BRNN) [14]

3) LSTM and BLSTM: Another disadvantage of classical RNN is the problem of exploding and vanishing gradients. As gradient descent is mostly used for training these neural networks, vanishing gradients lead to no learning at all and exploding gradients lead large oscillations of the objective function about the global minimum and results in convergence happening on local minima. To overcome this problem, long short-term memory (LSTM) structure was proposed [18]. The LSTM network is an upgraded version of RNN, in which the hidden layer is replaced by a more complex structure which has gates to remember crucial information and forget redundant information present in the data. The LSTM block is shown in the following figure 6: A classic LSTM block

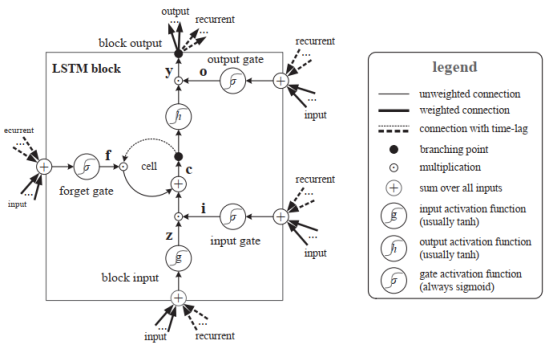

Fig. 6. A classical structure of long short-term memory (LSTM) block [17]

consists of three gates (input, forget and output gates), block input, a memory cell acting as a constant error carousel and an output activation function. Similar equations as that of BRNN can be written. LSTM computes the output vector sequence by iterating the following six equations:

$$\boldsymbol{z}_t = g(\boldsymbol{W}_z \boldsymbol{x}_t + \boldsymbol{R}_z \boldsymbol{y}_{t-1} + \boldsymbol{b}_z) \tag{12}$$

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{R}_i \boldsymbol{y}_{t-1} + \boldsymbol{b}_i) \tag{13}$$

$$\boldsymbol{z}_t = \sigma(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{R}_f \boldsymbol{y}_{t-1} + \boldsymbol{b}_f) \tag{14}$$

$$\boldsymbol{c}_t = \boldsymbol{z}_t \odot \boldsymbol{i}_t + \boldsymbol{c}_{t-1} \odot \boldsymbol{f}_t \tag{15}$$

$$\boldsymbol{o}_t = \sigma(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{R}_o \boldsymbol{y}_{t-1} + \boldsymbol{b}_o) \tag{16}$$

$$\boldsymbol{y}_t = h(\boldsymbol{c}_t) \odot \boldsymbol{o}_t \tag{17}$$

The LSTM network can be trained with standard Backpropagation Through Time (BPTT) algorithm [20]. The structure of bidirectional LSTM (BLSTM) is very similar to that given in figure 5 except that the hidden layer units is replaced by an LSTM block.

III. Summaries

A. ResNet-DNN based Channel Estimation and Equalization Scheme in FBMC/OQAM Systems

1) Architecture: The Res-DNN model consists of two 2-layers residual structures and two fully connected layers (FCL). IN FCL, every input node is connected to every output node. The major advantage of fully connected networks is

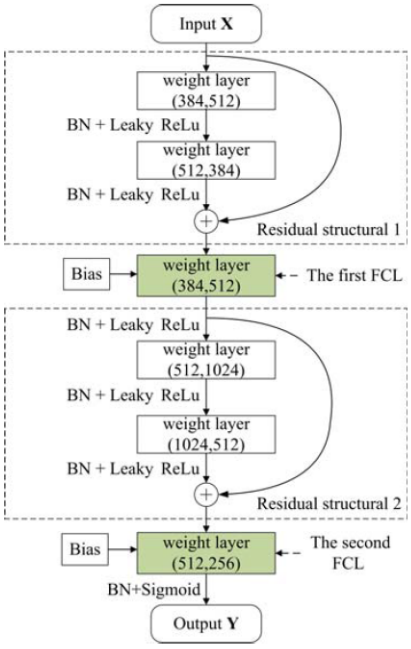

Fig. 7. The structure of proposed Res-DNN model [9]

that they are "structure agnostic" i.e. there are no special assumptions needed to be made about the input. While being structure agnostic makes fully connected networks very broadly applicable, such networks do tend to have weaker performance than special-purpose networks tuned to the structure of a problem space [17]. We have a FCL between the two residual structures and one FCL before the output layer. In addition, we add batch-normalization process between each layer so that data feedforward is more efficient in the networks. The activation function employed by the hidden layer is the Leaky Relu function, i.e., $\sigma(a) = max(0, a) + \alpha min(0, a)$, where α is a small constant, such as 0.2. But the activation function employed by the output layer is the sigmoid function, so that the output value can be controlled within the [0,1] interval.

2) Procedure: The process of generating training or test samples and corresponding labels are as follows:

1) A set of 4-QAM sequence and sent as a preamble.

2) Randomly bits as training labels which will be n-QAM modulated and sent over the channel.

3) The sequences generated in step 2 are successively passing through OQAM staggering, IFFT, filter bank, and signal superposition modules and corrupted with AWGN.

4) Transform the received signal by filtering (filter bank), FFT and OQAM de-staggering, then the real and imaginary parts of these complex sequences are interleaved and serialized into a long real-valued sequence. This long real-valued sequence is the training or testing sample we expect.

Repeating the operations of step 2 - step 4 for N times, we can get N training or testing samples and the corresponding labels. L2 loss function is used along with Adam optimizer to

reduce loss.

3) Results: Following is the performance of ResNet based channel equalisation:

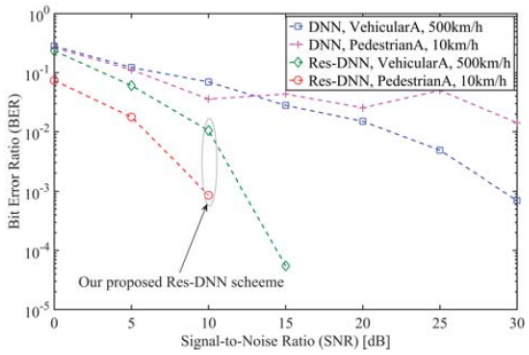

Fig. 8. The comparison of channel estimation performance between the proposed Res-DNN model and the conventional DNN model employed by the channel estimation scheme for OFDM systems

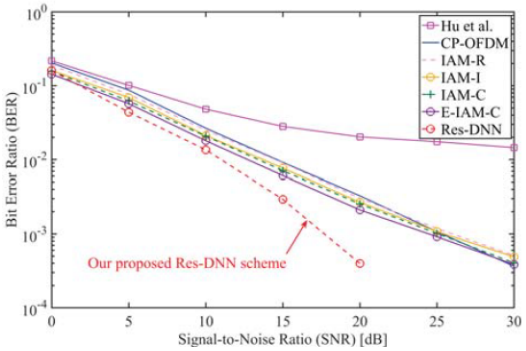

Fig. 9. The comparison of channel estimation performance between our proposed Res-DNN scheme and some improved IAM methods

We specifically note that under channel condition of Pedestrain-A with lowspeed (10km/h), the channel estimation performance of DNN based scheme is even worse than that of under the Vehicular- A channel with high-speed (500km/h). This is because the channel is quasi-static at low-speed, and the conventional DNN model is deep and no improvement measures are taken, so it is prone to over-fitting, which in turn leads to a degradation in performance.The Res-DNN based channel estimation scheme performs better.

B. Convolutional Recurrent Neural Network-based Channel Equalization

In this paper, an analytical formulation of channel equalization as a conditional probability distribution learning problem, which can be solved by a neural network model.Based on the formulation, a CRNN-based channel equalizer is proposed to cope with the problem of temporal variations of data as well as nonlinear channel distortions.

The cross-entropy loss function which measures the error of prediction that a given set of parameters $\mathbf{w}$ can result in is used for training. The function is given by

$$E(w) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{K}\mathbf{1}\left\{x^{(i)}=j\right\}\log\frac{e^{w_j^T y^{(i)}}}{\sum_{l=1}^{K}e^{w_l^T y^{(i)}}}\right]$$

where m is the total number of the training data $\left(y^{(i)}, x^{(i)}\right)$ and $\mathbf{1}\{\cdot\}$ is the indicator function, which means if the hypothesis made in the curly braces is correct, it returns 1 , otherwise it returns 0. In an instance of the training data, $x^{(i)}$(ith transmitted symbol) is defined as the label of $y^{(i)}$, which means the ground truth class of the symbol alphabet for $y^{(i)}$ Size of the symbol alphabet is K. the finite past of the received signal

$$y^{(i)} = \left[y^{(i)}, y^{(i-1)}, \ldots, y^{(i-N+1)}\right]^T$$

where N is the number of channel taps. Now that we can train a NN with an algorithm automatically tuning the parameters $\mathbf{w}$ to minimize the above error function.

1) Architecture: The first part is CNN, followed by RNN. It is a five layer network where the two Conv layers are used to learn matched filters, the dropout layer is used to avoid overfitting the training data, and the Max pooling layer is used to downsample the output of the previous layer.

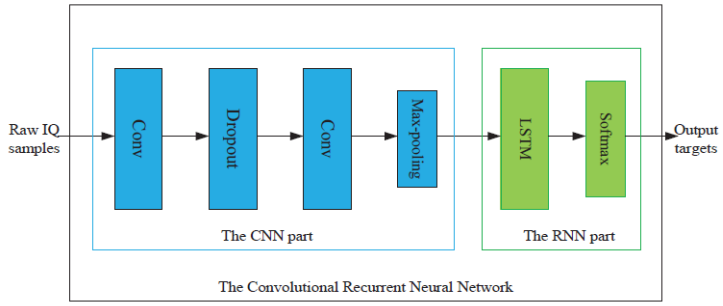

Fig. 10. CRNN architecture

Finally, we use Softmax activation function to derive the outputs for the last layer.We feed the learned filters into a subsequent layer for temporal modeling by utilizing LSTM cells, which specify the RNN part. The intuituion for using CNN and RNN parts is presented in the background information part already.

Thus the CRNN is trained to solve a K-class decision problem given tremendous (y(i), x(i)) instances known as training dataset.

2) Results: We first do hyperparameter optimization - it is the problem of choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a parameter whose value is used to control the learning process.

CRNNE PARAMETERS

Input layer size	2×12
Conv1 layer size	$2 \times 4 \times 64$
Conv2 layer size	$2 \times 4 \times 32$
Max-pooling size	2×2
LSTM layer size	100
Output layer size	4
Batch size	1024
SNR	15 dB

Fig. 11. Architecture parameters

The hyperparameters available are learning rate, dropout, and training epochs. In the paper, they have optimized the loss w.r.t. these three parameters using a subset of varying values. They got the best results for learning rate:= 10^{-3}, dropout = 0 (that means no overfitting was there) and training epochs = 40.
The SER performance is now evaluated using above parameters w.r.t RLS-based and MLP-based equalizers and we see that CRNN performs better than them.

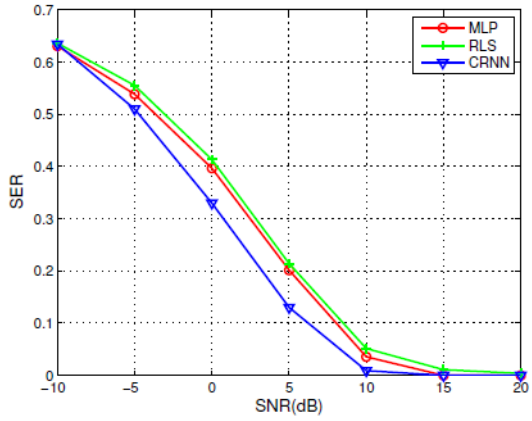

Fig. 12. SER performance of proposed CRNN versus RLS and MLP under different SNR.

C. BLSTM Channel Estimation Scheme

Unlike RNN, BLSTM can sense the influence of both the past and the future data on the current data being processed. This property of BLSTM can be related to the inter-symol interference (ISI) and inter-carrier interference (ICI) in the case of FBMC, as we know that the interference at TF point (m_0, n_0) is due to Ω_{m_0,n_0}. Because of the memory cells that are present, a BLSTM can estimate and compensate multipath time-varying channels better. One more motivation in using BLSTM is that the FBMC radio signals can be classified as time-series data and the frequency domain FBMC symbols are lattice distributed. They can be reshaped into vectors of certain length like text.

In contrast to the conventional channel estimation techniques, a deep BLSTM network is used as a function approximator to continually approximate the functions of the modules shown in blue background in figure 1. The flow chart of the implementation is shown in figure 13

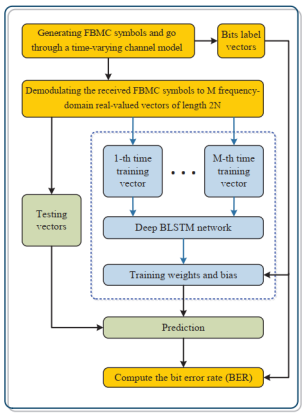

Fig. 13. The architecture of the proposed BLSTM-CE scheme [11]

The BLSTM employed is deep in both space and time. It has independent forward and backward hidden layers using which it can simultaneously take into account the interference from the past and the subsequent data sequence when predicting the current data sequence. Online learning method is used to train this network, and a sigmoid layer is used in the final layers so that the values of the predicted output vectors lie in the range $(0, 1)$. Mean-squared error loss function was used to train the network.
Numerical simulations were performed to compare and contrast the performance of this scheme with other schemes. The Vehicular-A and Pedestrian-A channel models released by the ITU [24] and the Jakes-Doppler sprectum set [23] were used. The results are shown in figures 14, 15.

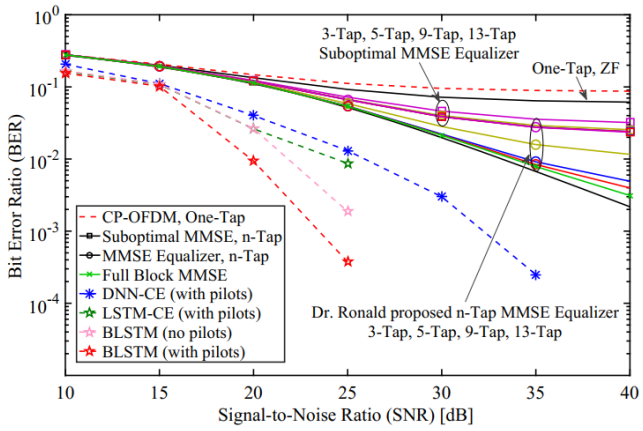

Fig. 14. The comparison of channel estimation and equalization performance between the proposed BLSTM-CE scheme and enhanced n-tap MMSE equalizer proposed in [23], and the DNN based scheme for FBMC system. The number of subcarriers N = 24, the number of symbols M = 15, 256-QAM modulation, *Hermite filter* is employed as the prototype filter and the overlapping factor O = 4, Vehicular-A channel with high-speed 500km/h.

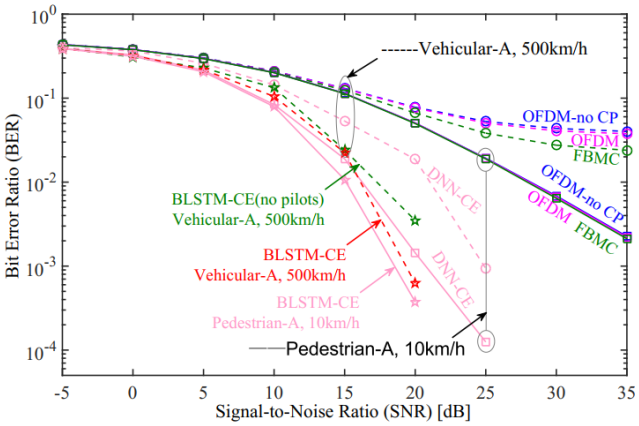

Fig. 15. The comparison of channel estimation performance between the proposed BLSTM-CE scheme and one-tap equalizer proposed in [25], and the DNN based scheme for the FBMC systems. The number of subcarriers N = 8, the number of symbols M = 5, 4-QAM modulation, *Hermite filter* is employed as the prototype filter and the overlapping factor O = 4.

IV. Extensions and Innovations

A. Extending for MIMO systems

In the CRNN and BLSTM papers only SISO was used, in CRNN single antenna pair was used. Here we propose to extend the deep learning based channel estimation schemes for the case of multi-input multi-output (MIMO) systems. A great advantage of any deep neural network is that it can be easily adapted to larger dimensions of input data by adjusting the depth of the hidden layers appropriately and the model learns the underlying task automatically. The same thing can be done here, by concatenating the signals received by the multiple receiver antennas on the receiver and a single deep learning model like CRNN or BLSTM can be used to estimate the effective MIMO channel directly.

B. Improvements to ResNet

In the ResNet architecture Paper, we used a simple ResNet. An improvement over this can be made by using architectures know as HighwayNets or DenseNets.

The Highway Network preserves the shortcuts introduced in the ResNet, but augments them with a learnable parameter to determine to what extent each layer should be a skip connection or a nonlinear connection. So, if in Resnet the output after a skip connection was F(x) + x, now it will be F(x) + αx [26].

The idea for DenseNet is that if connecting a skip connection from the previous layer improves performance, why not connect every layer to every other layer? That way there is always a direct route for the information backwards through the network. Furthermore we can make the weights learnable.

This architecture makes intuitive sense in both the feedforward and feed backward settings. In the feed-forward setting, a task may benefit from being able to get low-level feature activations in addition to high level feature activations.

The use of HighwayNet or DenseNet will most likely improve performance over the ResNet

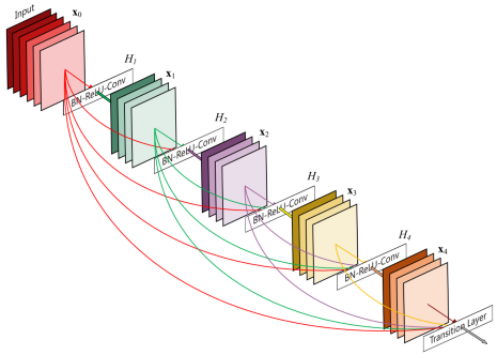

Fig. 16. DenseNet architecture

C. Model Scaling and EfficientNet

In all three papers, they are have limited number of layers, 4 in BLSTM, 2 convolutional layers architecture, and 2 residual structures in the ResNet paper. We think that increasing the depth of all three architectures i.e Model scaling will give better performance - this however will come at increased resource cost.

Model scaling is about scaling the existing model in terms of model depth, model width, and less popular input image resolution to improve the performance of the model. Depth wise scaling is most popular amongst all, e.g. ResNet can be scaled from Resnet18 to ResNet200. Here ResNet10 has 18 residual blocks and can be scaled for depth to have 200 residual blocks. ResNet200 delivers better performance than ResNet18 and thus, manually scaling works pretty well. But there is one problem with traditional manual scaling method, after a certain level, scaling doesn't improve performance. It starts to affect adversely by degrading performance.

So we propose that with the EfficientNet scaling technique (paper referenced in [28]) be used to do compound scaling. Compound scaling suggests that instead of scaling only one model attribute out of depth, width, and resolution; strategically scaling all three of them together delivers better results. We strongly think that model scaling using the EfficientNet technique will improve performance.

D. Graph Neural Networks

We saw three architectures used for channel estimation in this reflection paper. We propose that an architecture which can be used for channel estimation is Graph Neural networks (GNN). Graph neural networks refer to the neural network architectures that operate on a graph. The aim of a GNN is for each node in the graph to learn an embedding containing information about its neighborhood (nodes directly connected to the target node via edges). This embedding can then be used for different problems like node labelling, node prediction, edge prediction, etc. [29]. This architecture has a possibility of performing well due to its structured approach.

E. Zero-Shot and Few-Shot Learning

For the ultra-low latency situations, techniques like zero-shot learning and few-shot learning can be used to train the

deep learning model for channel estimation and equalization. As the name suggests, these techniques require very less amount of data than a standard deep neural network and give very similar performance.

F. TinyML

All the experiments and simulations that were mentioned above are performed on a computer. To realize these schemes practically one needs to find a low-cost and less-complex way to implement these schemes. In recent times, the concept of TinyML is getting popular wherein the machine learning techniques are optimized to use along with ultra-low-power embedded IoT devices. Google has open-sourced the Tensorflow Lite (Micro) [28] very recently, using which one can deploy the standard ML models on small microprocessors. Using this we can evaluate and benchmark different schemes using standard microprocessors/microcontrollers like Arduino.

V. Conclusion

In this reflection paper we gave the background information and summary of the papers "A ResNet-DNN based Channel Estimation and Equalization Scheme in FBMC/OQAM Systems" [9], "Convolutional recurrent neural network-based channel equalization: An experimental study" [10] and "Channel Estimation and Equalization Based on Deep BLSTM for FBMC-OQAM Systems" [11]. We then saw the extensions and innovations possible with respect to these papers proposed by us. The references and a summary about them is given below.

References

[1] J-P. Javaudin, D. Lacroix, and A. Rouxel, "Pilot-aided channel estimation for OFDM/OQAM," in VTC Spring, pp. 1581-1585, Apr. 2003.

[2] C. Lele, J.-P. Javaudin, R. Legouable, A. Skrzypczak and P. Siohan, "Channel estimation methods for preamble-based OFDM/OQAM modulations," in Proc. Eur. Wireless Conf. (EW), pp. 1-7, Paris, France, Apr. 2007.

[3] D. Lacroix-Penther and J-P. Javaudin, "A new channel estimation method for OFDM/OQAM," in Proc. Int. OFDM Workshop., pp.1-5, Sept. 2002.

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

[5] X. Cheng, D. Liu, C. Wang, S. Yan and Z. Zhu, "Deep Learning based Channel Estimation and Equalization Scheme for FBMC/OQAM Systems," IEEE Wireless Commun. Lett. vol. 8, no. 3, 2019.

[6] C. K. Wen, W. T. Shih and S. Jin, "Deep Learning for Massive MIMO CSI Feedback," IEEE Wireless Commun. Lett., vol. 7, no. 5, pp. 748-751, Oct. 2018.

[7] M. Kim, W. Lee, and D. H. Cho, "A Novel PAPR Reduction Scheme for OFDM System based on Deep Learning," IEEE Commun. Lett., vol. 22, no. 3, PP. 510-513, Mar. 2018.

[8] H. Ye, G. Y. Li, and B. H. F. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems," IEEE Wireless Commun. Lett., vol. 7, no. 1, pp. 114-117, Feb. 2018.

[9] X. Cheng, D. Liu, Z. Zhu, W. Shi and Y. Li, "A ResNet-DNN based Channel Estimation and Equalization Scheme in FBMC/OQAM Systems," in Proc. Int. Conf. Wireless Commun. and Signal Process. (WCSP), pp. 1-5, Hangzhou, China, Oct. 2018.

[10] Y. Li, M. Chen, Y. Yang, M. Zhou, and C. Wang, "Convolutional recurrent neural network-based channel equalization: An experimental study," in Proc. 23rd Asia-Pacific Conf. Commun. (APCC), pp. 1-6, Perth, WA, Dec. 2017.

[11] X. Cheng, D. Liu, S. Yan, W. Shi and Y. Zhao, "Channel Estimation and Equalization Based on Deep BLSTM for FBMC-OQAM Systems," ICC 2019 - 2019 IEEE International Conference on Communications (ICC), Shanghai, China, 2019, pp. 1-6, doi: 10.1109/ICC.2019.8761647.

[12] Introduction to Residual Networks: https://www.geeksforgeeks.org/introduction-to-residual-networks/

[13] Vanishing gradient problem: https://en.wikipedia.org/wiki/Vanishing_gradient_problem

[14] A. Graves, A. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural networks," in IEEE Proc. Int. Conf. Acoustics, Speech and Signal Process. (ICASSP), pp. 6645-6649, Vancouver, BC, 2013.

[15] CRNN:https://www.analyticsvidhya.com/blog/2020/11/a-short-intuitive-explanation-of-convolutional-recurrent-neural-networks/

[16] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," IEEE Tran. Signal Process, vol. 45, no. 11, Nov. 1997.

[17] K. Greff, R. K. Srivastava, J. Koutnk, B. R. Steunebrink and J. Schmidhuber, "LSTM: A Search Space Odyssey," IEEE Trans. Neural Networks and Learn. Sys., vol. 28, no. 10, pp. 2222-2232, Oct. 2017.

[18] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 15 Nov. 1997.

[19] CRNN image:https://raw.githubusercontent.com/keunwoochoi/music-auto_tagging-keras/master/imgs/diagrams.png

[20] A. Graves and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM and other neural network architectures," Neural Networks, vol. 18, no. 5-6, pp. 602-610, 2005.

[21] E. Kofidis, D. Katselis, A. Rontogiannis, and S. Theodoridis. "Preamble-based channel estimation in OFDM/OQAM systems: A review," Signal Process., vol. 93, no. 7, pp. 2038-2054, Jul. 2013.

[22] J. Du and S. Signell, "Time Frequency localization of pulse shaping filter in OFDM/OQAM systems," in Proc. Int. Conf. Inf., Commun. Signal Process. (ICICS), pp. 15, Dec. 2007.

[23] R. Nissel, M. Rupp, and R. Marsalek, "FBMC-OQAM in Doubly Selective Channels: A New Perspective on MMSE Equalization," in Proc. IEEE Int. Workshops on Signal Process. Advances in Wireless Commun. (SPAWC), pp. 1-5, Sapporo, Japan, Jul. 2017.

[24] ITU: Guidelines for evaluation of radio transmission technologies for IMT-2000, Recommendation ITU-R M.1225, 1997 (Table 5).

[25] R. Nissel and M. Rupp, "OFDM and FBMC-OQAM in Doubly-Selective Channels: Calculating the Bit Error Probability," IEEE Commun. Lett., vol. 21, no. 6, pp. 1297-1300, Jun. 2017.

[26] HighwayNets and DenseNets: https://www.kdnuggets.com/2016/12/resnets-highwaynets-densenets-oh-my.html

[27] EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks https://arxiv.org/pdf/1905.11946.pdf

[28] David, R., "TensorFlow Lite Micro: Embedded Machine Learning on TinyML Systems", arXiv e-prints, 2020.

[29] Graph Neural Networks: https://heartbeat.fritz.ai/introduction-to-graph-neural-networks-c5a9f4aa9e99

VI. Author Details and Brief Background

Amol Girish Shah is a junior undergraduate pursuing Dual Degree with specialization in Communication and Signal Processing from the Department of Electrical Engineering at Indian Institute of Technology Bombay, Mumbai. He will be receiving his BTech + MTech degree by May 2023. His research interests lie broadly in the field of information technology specifically in signal processing and machine learning.

Mantri Krishna Sri Ipsit is a junior undergraduate pursuing Bachelor of Technology from the Department of Electrical Engineering at Indian Institute of Technology Bombay, Mumbai. He will be receiving his BTech degree by May 2022. His research interests lie broadly in the field of information technology specifically in signal processing and machine learning.