Design and Prototyping of Neural Network Compression for Non-Orthogonal IoT Signals

Tongyang Xu and Izzat Darwazeh

Department of Electronic and Electrical Engineering, University College London, London, UK
Email: tongyang.xu.11@ucl.ac.uk, i.darwazeh@ucl.ac.uk

***Abstract*—The non-orthogonal IoT signal, following the bandwidth compression spectrally efficient frequency division multiplexing (SEFDM) characteristics, can bring benefits in enhanced massive device connections, signal coverage extension and data rate increase, but at the cost of computational complexity. Resource-constrained IoT devices have limited memory storage and complex signal processing is not allowed. Machine learning can simplify signal detection by training a general data-driven signal detection model. However, fully connected neural networks would introduce processing latency and extra power consumption. Therefore, the motivation of this work is to investigate different neural network compression schemes for system simplification. Three compression strategies are studied including topology compression, weight compression and quantization compression. These methods show efficient neural network compression with trade-offs between computational complexity and bit error rate (BER) performance. Practical neural network prototyping is evaluated as well on a software defined radio (SDR) platform. Results show that the practical weight compression neural network can achieve similar performance as the fully connected neural network but with great resource saving.**

***Index Terms*—Neural network, machine learning, neural network compression, Internet of things, non-orthogonal, spectral efficiency, software defined radio, prototyping.**

I. Introduction

Machine learning has greatly influenced our lives in various intelligent applications. Recently, machine learning is stepping into signal communications due to its abilities in dealing with complex system architecture and impossible mathematical modelling. In wireless communications, the work in [1] explained the potential applications of deep learning in physical layer and later extended to specific multiple input multiple output (MIMO) [2] and orthogonal frequency division multiplexing (OFDM) [3]. The autoencoder concept was used in MIMO systems, which can emulate encoder and decoder at the same time. In this case, the entire system is a black box, which can be globally optimized. Recent progress in work [4] showed a realistic over-the-air experiment via using autoencoder to effectively train the entire physical layer.

In next generation internet of things (IoT) [5], power and spectral efficiency are the focuses. Therefore, the non-orthogonal signal waveform SEFDM [6], [7], showing improved power and spectral efficiency, is being developed. Unlike the typical OFDM signal waveform, SEFDM introduces self-created inter carrier interference (ICI) and therefore complex signal detection. Previous work [7] focused on data rate improvement based on the QPSK modulation format. Since it is for uplink communications, a powerful but complex sphere decoding (SD) detector can be used. However, for downlink channels, this solution is not practical because of the limited battery life in each IoT device.

Simplification of signal processing, relying on specially designed neural networks, in each IoT device is the motivation of this work. A fully connected neural network is first designed and studied. Moreover, according to the non-orthogonal signal waveform characteristic, different neural network compression strategies are investigated to simplify further the IoT device complexity. Neural network compression is necessary for practical hardware implementation. There are many compression strategies being summarized in [8], [9]. First, as explained in [9], activation functions may have different operation complexities but at the cost of variable performance. Second, weight pruning and quantization optimization are evaluated in [10]. Furthermore, work in [11] proposed HashedNets, which uses a hash bucket to constrain a group of neural network connections to a single parameter. Then backpropagation is used to fine tune the assigned parameter. Moreover, work in [12] designed BinaryNet and showed a 1-bit arithmetic operation for both training and inference stages.

In this work, taking into account the SEFDM waveform characteristic, three neural network compression schemes are evaluated theoretically and practically. The first one is topology compression via designing efficient network neuron connections. Second, weight compression is to save memory storage and computation cost via truncating unimportant weights. Third, quantization compression is to lower the precision of each weight at the cost of accuracy. In addition, practical neural network prototyping is implemented to verify the neural network weight compression.

II. Signal Waveform Principle

In IoT communications, signals are simply designed using low order modulation formats and a small number of sub-carriers for power saving purposes. The non-orthogonal signal SEFDM brings advantages such as bandwidth saving as shown in Fig. 1. The basic idea is to pack sub-carriers closer leading to improved spectral efficiency at the cost of violating the OFDM orthogonal property. The detailed signal model is referred to [6], [7]. Assume N is the number of sub-carriers, k

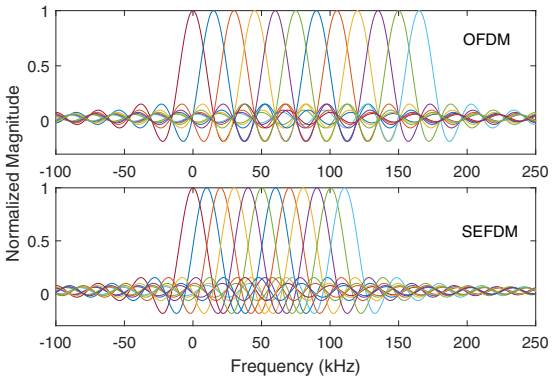

Fig. 1. Sub-carrier packing schemes for different multicarrier signals. OFDM (12 sub-carriers, bandwidth is B). SEFDM (12 sub-carriers, bandwidth compression factor α, bandwidth is $\alpha \times B$).

is the time sample index and α is the bandwidth compression factor. The self-created ICI is mathematically defined as

$$c_{m,n} = \frac{1}{N}\sum_{k=1}^{N} e^{\frac{j2\pi mk\alpha}{N}} e^{-\frac{j2\pi nk\alpha}{N}} = \begin{cases} 1 & , \ m = n \\ \frac{1-e^{j2\pi\alpha(m-n)}}{N(1-e^{\frac{j2\pi\alpha(m-n)}{N}})} & , \ m \neq n \end{cases}, \quad (1)$$

where the term $e^{\frac{j2\pi nk\alpha}{N}}$ is the modulation matrix while its conjugate term $e^{\frac{-j2\pi nk\alpha}{N}}$ is demodulation matrix. Therefore, the multiplication in (1) represents the correlation between two arbitrary sub-carriers associated with m and n indices. It is noted that the cross correlation terms ($m \neq n$), indicating the self-created ICI, are not zero when $\alpha < 1$. Powerful signal detectors have to be applied to remove the ICI but with high computational complexity.

III. Fully Connected Neural Network

In OFDM communications, neural networks are designed to improve channel estimation and equalization accuracy [3]. Sub-carriers in OFDM are orthogonally packed and ICI does not exist. This brings benefits to the neural network design such as independent and parallel sub-nets architecture, which can speed up neural networks training. However, unlike OFDM signals, ICI exists in SEFDM and all the neurons have to be connected to explicitly model the interference. Thus, a fully connected neural network architecture is employed and demonstrated in Fig. 2.

It should be noted that instead of the autoencoder training for entire communication systems in [1], [4], the aim of this work is to train a neural network specifically for signal detection. In addition, this work focuses on narrowband applications in IoT. Therefore, we consider four sub-carriers as a sub-net in the designed neural network, which is sufficient for IoT applications. For a large number of sub-carriers, a multi-band architecture, including multiple sub-nets, are to be considered.

The trained full connection-deep neural network (F-DNN) [13] has one input layer, two hidden layers and one output layer. The number of neurons at each layer are 4, 14, 14, 4. Sigmoid activation function is used at each layer. The basic operations of this neural network are multiplication and addition and the computation complexity is proportional to the number of layers and the number of neurons at each layer.

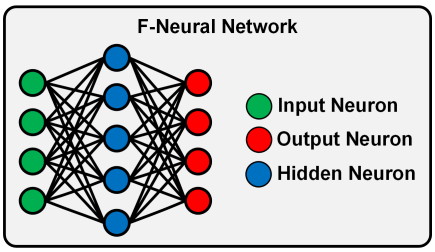

Fig. 2. Fully connected neural network topology for SEFDM signals. For simplicity, only one hidden layer is illustrated. For deep learning scenarios, more hidden layers are required.

IV. Neural Network Compression

In this work, taking into account the employed waveform characteristics, we consider neural network simplification schemes such as network topology compression, weight compression, and quantization compression.

A. Topology Compression

The fully connected neural network is widely used since all the neurons are connected leading to an accurate network modelling. However, due to the fully connected architecture, signal processing is complex since all the neurons have to be processed simultaneously. In this section, we proposed to optimize signal processing using different neural network topologies according to the SEFDM waveform characteristic. In this case, neurons are partially connected and can be processed in parallel.

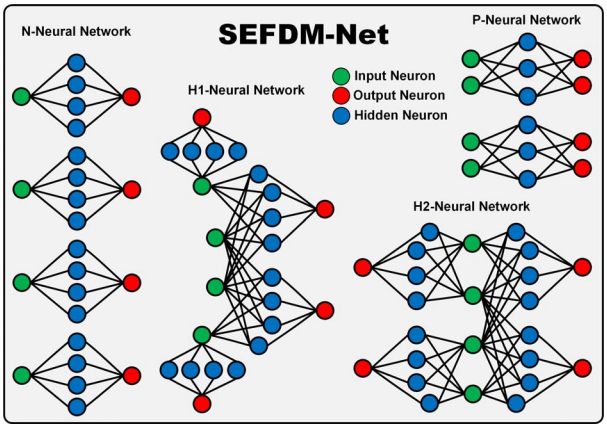

Fig. 3. Flexible neural network topologies for SEFDM signals. For simplicity, only one hidden layer is illustrated. For deep learning scenarios, more hidden layers may be required.

Due to the self-created ICI within SEFDM signals, neuron connections can be optimally designed according to the interference distribution among sub-carriers. Four neural network

topologies [13] are designed and compared in Fig. 3. The first one is termed no connection-neural network (N-NN), in which each input neuron is independently modelled and a parallel architecture is derived. The benefit of this network structure is its parallel processing. The second network is partial connection-neural network (P-NN), which connects adjacent two input neurons leading to two independent sub-nets. This network topology considers extra interference from one adjacent neuron and would lead to more accurate network modelling. In order to improve further the modelling accuracy, two hybrid connection neural networks, hybrid1-neural network (H1-NN) and hybrid2-neural network (H2-NN), are designed. The main improvement of the two networks is comprehensive interference modelling. In H1-NN, the middle two input neurons are modelled together considering interference from adjacent neurons. In H2-NN, one extra improvement step is its edge neuron interference modelling.

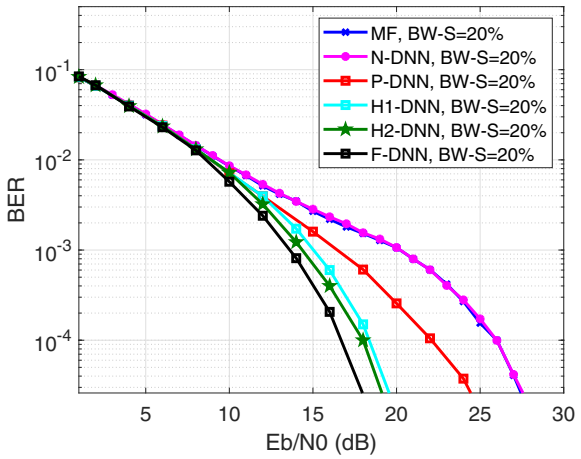

Fig. 4. BER performance of SEFDM-Net on SEFDM signals of 20% bandwidth saving (α=0.8).

The performance of the four neural network topologies is compared in Fig. 4 where two hidden layers are used. It is clearly seen that the F-DNN shows the best performance, which outperforms matched filter (MF) by 9 dB at BER=10^{-4}, due to the connections of all the neurons and therefore a more accurate network modelling. For other proposed neural networks, the performance is worse. The no connection-deep neural network (N-DNN) shows the worst performance, which has the same performance as the MF. For other neural networks, due to different neuron connections and therefore variable interference modelling accuracy, performance is various but better than N-DNN.

Table I: Number of weights in each neural network.

Parameters	N-DNN	P-DNN	H1-DNN	H2-DNN	F-DNN
No. of Weights	448	504	448 560	504 560	616

The complexity of each neural network is computed based on the number of weights (i.e. the number of neuron connections). Reusable architectures are considered for the neural networks in Fig. 3. The resource occupation of each neural network, reflected by the number of weights in each sub-net, is numerically summarized in Table I. The fully connected neural network needs 616 weights while the remaining neural networks require fewer weights for each reuse. This indicates that the proposed topology compression neural networks have simpler system designs than the fully connected neural network. However, it is at the cost of processing speed due to the reuse architecture. It should be noted that the hybrid neural networks have two independent sub-nets leading to variable number of weights in each sub-net.

B. Weight Compression

The weight compression aims to locate redundancy in the model and remove unimportant weights [10] and corresponding neuron connections while maintaining reasonable performance. It can optimize the neural network structure to a sparse network and reduce the network complexity.

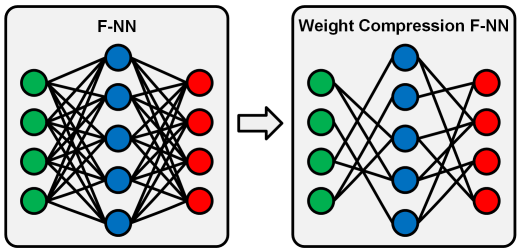

Fig. 5. Weight compression illustration in a fully connected neural network. For simplicity, only one hidden layer is illustrated. For deep learning scenarios, more hidden layers are required.

The effective connection of neurons within the SEFDM-Net is determined by weights. The weight of each neuron connection is not equally assigned, which gives us inspiration that small weights can be truncated and further cut the neuron connection. Therefore, a compressed neural network with reduced neuron connections is obtained. Depending on a predefined truncation weight threshold (i.e. an absolute magnitude value), all weights with absolute magnitudes below the threshold will be removed together with corresponding neuron connections.

Table II: Simulation trained WC-DNN model.

Truncation Weight Threshold	Total Weights	Reserved Weights	Resource Compression	Improved Processing Speed
0.6	616	463	24.8%	33.0%
0.8	616	406	34.1%	51.7%
1	616	355	42.4%	73.5%
1.5	616	266	56.8%	131.6%

Based on the studies in Section IV-A, the fully connected neural network achieves the best performance. Therefore, we

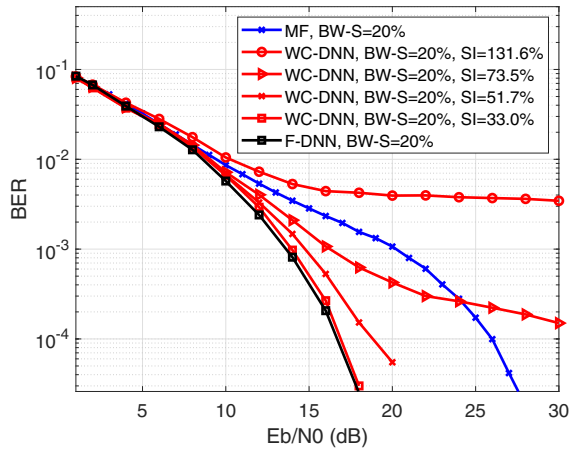

Fig. 6. BER performance of weight compression on SEFDM signals of 20% bandwidth saving (α=0.8).

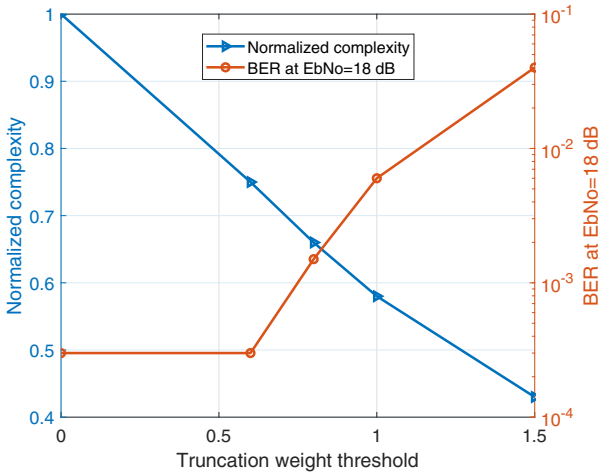

Fig. 7. Trade-off in complexity and BER versus truncation weight threshold in WC-DNN.

focus on the weight compression in fully connected neural networks, which is termed weight compression-DNN (WC-DNN). Table II summarizes the amount of resources consumed under different truncation weight thresholds. With the increase of the truncation weight threshold, the amount of reserved weights are reduced and therefore improved processing speed. However, the trade-off is the bit error rate (BER) performance. As is shown in Fig. 6, with more weights truncated, the BER performance becomes worse. For the 42.4% and 56.8% resource compression, error floors start to appear. For the 34.1% resource compression, the performance gap becomes narrow. For the 24.8% resource compression, the performance is identical to the original fully connected neural network. Therefore, to trade off complexity and performance, the 24.8% neural network compression is optimal. For special scenarios such as complexity-driven applications, the 34.1% resource compression would be preferred. The performance loss could be compensated using powerful channel coding. It should be noted that below Eb/N0=24 dB, the neural network with 42.4% resource compression outperforms the typical MF but with 73.5% processing speed improvement.

The trade-off of performance and complexity for the WC-DNN based system is illustrated in Fig. 7. Normalized complexity is calculated and the full complexity with '0' truncation weight threshold is assumed to be 100%. With the increase of the truncation threshold, BER becomes worse while complexity is decreased gradually. It is inferred that cutting more neuron connections affects greatly the system modelling accuracy but with reduced complexity.

C. Quantization Compression

In hardware implementation, parameters are configured in fixed-point representations for the purpose of hardware resource saving. Unlike floating-point parameters, fixed-point parameters have lower precision, where the decimal part is determined by a fixed number of bits.

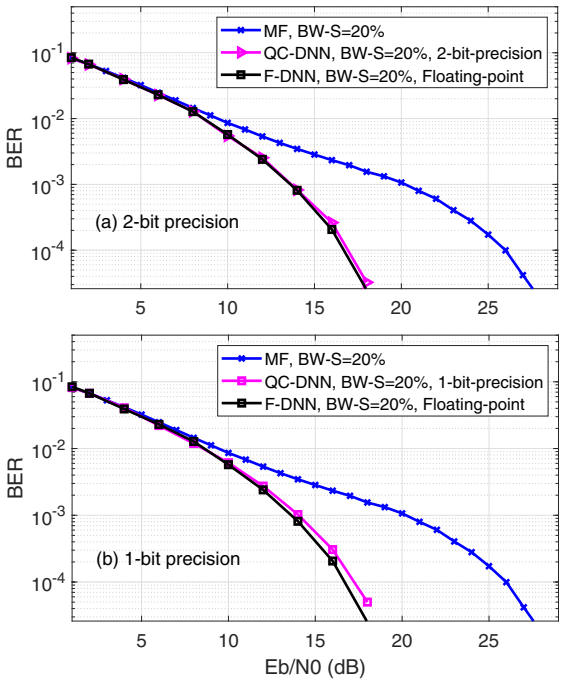

Fig. 8. BER performance of quantization compression on SEFDM signals of 20% bandwidth saving (α=0.8).

The precision in this work is defined as the number of bits represented for the decimal part in each weight. We test two scenarios in Fig. 8. The aim of quantization compression DNN (QC-DNN) is to lower the resolution of each weight to an accepted level and would not affect BER performance. Results in Fig. 8(a) show that the 2-bit precision quantization compressed neural network can achieve the same performance as the original fully connected neural network. By further compressing the precision to 1-bit in Fig. 8(b), the quanti-

zation compressed neural network still presents very close performance as the F-DNN with neglect performance loss. This gives us a conclusion that the SEFDM neural network is not sensitive to quantization compression and this discovery paves the way to simple hardware implementation.

V. Neural Network Prototyping

The neural network prototyping is based on a software defined radio (SDR) platform [14], which is shown in Fig. 9. It has a complete radio frequency (RF) chain and its digital function is realized in software. RF 3026C is the signal generator [15], which can support up to 14-bit complex symbols. RF 3035C [16] is used to receive and convert analog signals to digital signals. The transmitter and the receiver are frequency synchronized using RF 3011C [17]. However, due to signal transmission delay, received signals are timing offset by a few samples. In addition, the testbed has local oscillator (LO) phase offset and sampling phase offset. All of these effects have to be taken into account in the practical neural network training stage. Moreover, we expect signals of a small number of sub-carriers at low sampling frequency since this work is focusing on indoor narrowband IoT applications. The narrow bandwidth of IoT signals brings better channel conditions than that in wide band signals. Therefore, in this work, we would expect time-invariant frequency flat channels with additive white Gaussian noise (AWGN) and potential equipment distortions.

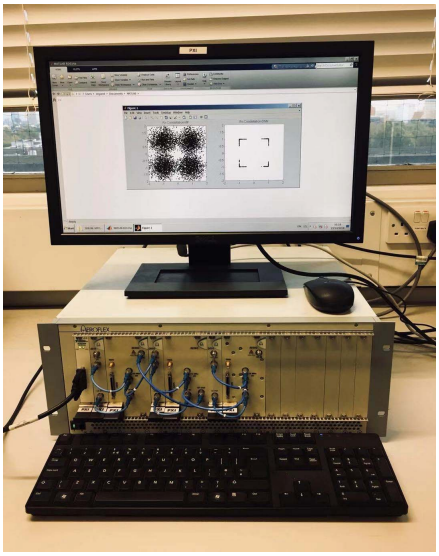

Fig. 9. Neural network prototyping on a SDR platform.

In the autoencoder work [4], stable RF effects are assumed to simplify the network training process. However, in practical scenarios, those effects are random and cannot be deterministically modelled. Work in [4] employs a neural network for an entire communication system where traditional compensation algorithms are not allowed. Therefore, a two-phase training strategy has to be used in their model where the first stage makes use of constant parameters for training and the second stage fine tunes the trained neural network based on practical data. In our work, we do not have such a problem since we are merely focusing on signal detection neural network design and the random RF effects can be dealt with using typical channel compensation methods.

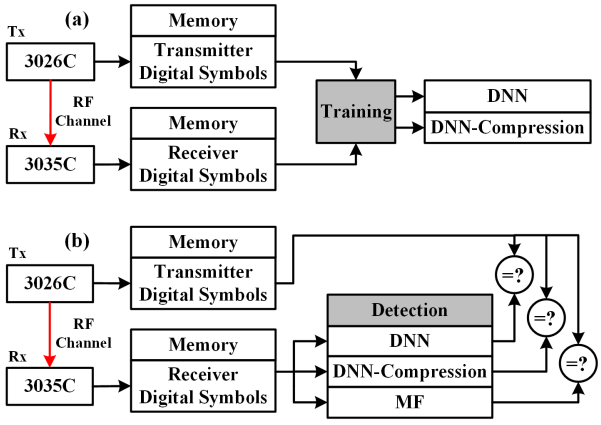

Fig. 10. Block diagram of neural network training and detection.

To avoid simulation uncertainty, this work employs a SDR-training strategy as shown in Fig. 10(a). The RF 3026C generates random bits in 1.92 MS/s sampling rate at 2.4 GHz carrier frequency. The bit stream is mapped to 4QAM symbols which are later modulated on four sub-carriers. One copy of the modulated symbol is sent to a memory for storage. The other one is delivered to the receiver RF 3035C via an RF channel. The digitized data is saved to a memory for training. To guarantee a fair comparison, the fully connected DNN and compressed DNN systems are trained based on the same transmission and reception data set. For the inference/detection stage as shown in Fig. 10(b), the SDR testbed transmitter repeats the same operation such as data saving and data delivery. At the receiver, the digitized data interfered by ICI is fed to three systems for signal detection. The detected symbols are compared with the memory stored symbols for BER.

Table III: Experiment trained WC-DNN model.

Truncation Weight Threshold	Total Weights	Reserved Weights	Resource Compression	Improved Processing Speed
0.6	616	469	23.9%	31.3%

We test the weight compression neural network in the SDR testbed and compare its performance with the fully connected neural network and the MF system. We select the truncation weight threshold 0.6 and summarize the statistics of the WC-DNN system in Table III. Comparing to Table II, it is clearly seen that minor difference exists such as the number of reserved weights, resource compression ratio and improved processing speed. The mismatch may comes from limited hardware resolution and unexpected hardware imperfections. It is inferred that the simulation trained model may not be directly used in practical systems. To accurately

train a practical model, experiment collected data would be preferred for the training.

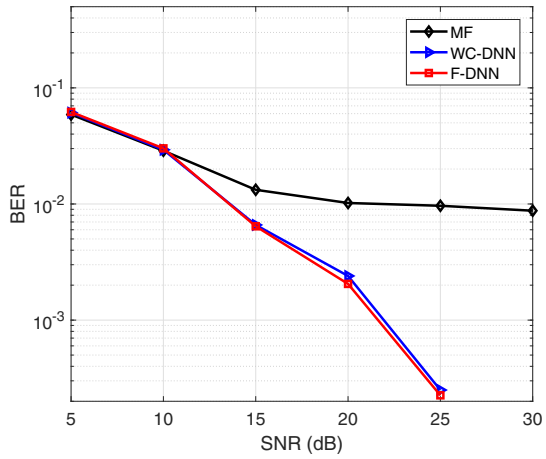

Fig. 11. Practical BER performance of weight compression.

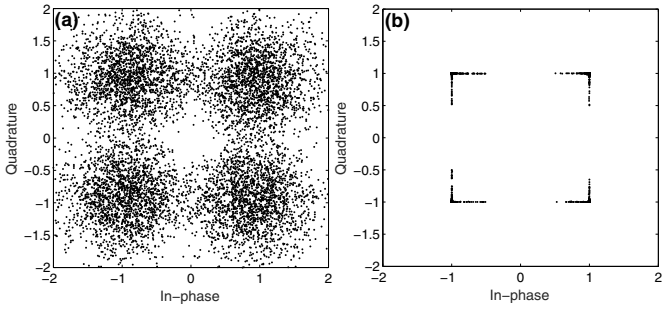

Fig. 12. Constellation of (a) MF results and (b) WC-DNN results.

Fig. 11 presents the results for WC-DNN following practical SDR-training and SDR-detection strategies. It is clearly seen that the curves show similar trend to that in Fig. 6. The MF shows error floor because of strong signal ICI. The F-DNN shows much better performance than MF. The WC-DNN achieves similar performance as the F-DNN but with 23.9% resource compression.

Fig. 12(a) represents original SEFDM constellation points at SNR=15 dB. The constellation is scattered by both AWGN and ICI. Fig. 12(b) shows WC-DNN neural network recovered constellation with a constrained boundary at '-1' and '1'. This is due to the use of Sigmoid function at the final neural network layer. It is inferred that due to the restricted boundary, the waveform self-created ICI is mitigated to some degree.

VI. Conclusions

This work aims to simplify neural network design for a non-orthogonal IoT signal via network compression. A fully connected neural network achieves the best performance but at the cost of computational complexity. Therefore, three neural network compression strategies were investigated. First, topology compression leads to different network structures with lower complexity than the fully connected neural network at the cost of performance. Second, weight compression truncates unimportant neuron connections in a network to simplify the system architecture. Simulation results showed 24.8% complexity reduction with no performance loss. Third, quantization compression is beneficial to hardware implementation. Studies in this work showed that the SEFDM system performance is not sensitive to the quantization effect and even 1-bit precision can achieve reasonable performance. In addition to simulation modelling, practical neural network prototyping was investigated in a software defined radio SDR platform. Experiment results demonstrated that the weight compressed neural network WC-DNN achieved similar performance as the fully connected F-DNN but with 23.9% complexity reduction.

References

[1] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563–575, Dec 2017.

[2] T. O'Shea, T. Erpek, and T. C. Clancy, "Deep learning based MIMO communications," *ArXiv e-prints*, Jul. 2017.

[3] H. Ye, G. Y. Li, and B. H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Communications Letters*, vol. 7, no. 1, pp. 114–117, Feb 2018.

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

[5] J. Xu, J. Yao, L. Wang, Z. Ming, K. Wu, and L. Chen, "Narrowband Internet of Things: Evolutions, technologies and open issues," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 1449–1462, June 2018.

[6] T. Xu, C. Masouros, and I. Darwazeh, "Waveform and space precoding for next generation downlink narrowband IoT," *IEEE Internet of Things Journal*, 2019 (to appear).

[7] T. Xu and I. Darwazeh, "Uplink narrowband IoT data rate improvement: Dense modulation formats or non-orthogonal signal waveforms?" in *2018 IEEE 29th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, Bologna, Italy, Sep. 2018.

[8] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "A survey of model compression and acceleration for deep neural networks," *ArXiv e-prints*, Dec. 2017.

[9] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018.

[10] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and Huffman coding," *ArXiv e-prints*, Feb. 2015.

[11] W. Chen, J. T. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen, "Compressing neural networks with the hashing trick," *CoRR*, vol. abs/1504.04788, Apr. 2015.

[12] M. Courbariaux and Y. Bengio, "Binarynet: Training deep neural networks with weights and activations constrained to +1 or -1," *ArXiv e-prints*, Mar. 2016.

[13] T. Xu, T. Xu, and I. Darwazeh, "Deep learning for interference cancellation in non-orthogonal signal based optical communication systems," in *2018 Progress In Electromagnetics Research Symposium - Spring (PIERS)*, August 2018 (invited).

[14] T. Xu and I. Darwazeh, "Non-orthogonal narrowband Internet of Things: A design for saving bandwidth and doubling the number of connected devices," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2120–2129, June 2018.

[15] Aeroflex Test Solutions, "3020 Series RF Signal Generators," http://ats.aeroflex.com/modular-instrumentation-pxi-products/pxi-systems-solutions/modules/3020-series-rf-signal-generators, May 2017.

[16] ——, "3030 Series PXI RF Digitizers," http://ats.aeroflex.com/modular-instrumentation-pxi-products/pxi-systems-solutions/modules/3030-series-pxi-rf-digitizers, May 2017.

[17] ——, "3010/3011 RF Synthesizers," http://ats.aeroflex.com/modular-instrumentation-pxi-products/pxi-systems-solutions/modules/3010-3011-rf-synthesizers, May 2017.