Deep Learning Based Beamforming Neural Networks in Downlink MISO Systems

Wenchao Xia†,¶ Gan Zheng§, Yongxu Zhu§, Jun Zhang†,¶ Jiangzhou Wang‡, and Athina P. Petropulu£

Abstract—**Beamforming techniques play an important role in multi-antenna communication systems and this work focuses on the downlink power minimization problem under a set of quality of service constraints. Conventional iterative algorithms can obtain optimal solutions but at the cost of high computational delay. Fast beamforming can be achieved by leveraging the powerful deep learning techniques. In this work, we propose a beamforming neural network (BNN), based on convolutional neural networks and exploitation of expert knowledge, for the power minimization problem. Instead of estimating beamforming matrix directly, we predict key features using the BNN which takes complex channel as input. Then the beamforming matrix is recovered from the predictions according to the uplink-downlink duality. The BNN adopts the supervised learning method with a loss function based on the mean-squared error metric to update network parameters. Simulation results show the BNN can achieve satisfactory performance with low computational delay.**

I. Introduction

The beamforming technique, playing a key role in multi-antenna systems, has attracted much attention for its ability to realize the performance gain. One typical application scenario is the power minimization problem under a set of quality of service (QoS) constraints [1–5]. The power minimization problem can be reformulated as a second-order cone programming (SOCP) [1, 2] or semidefinite programming (SDP) problem [3], which can be solved directly by an optimization software package such as CVX. Its optimal solution can also be obtained using iterative algorithms such as Algorithm A of [4] and the dual algorithm of [5]. The main drawbacks of existing iterative algorithms are the high computational complexity and delay. As a result, the beamforming technique is unable to meet the demands of real-time applications in the fifth-generation (5G) system and beyond, such as autonomous vehicles and mission critical communications. Even in non-real-time applications, where the small-scale fading varies in the order of milliseconds[6], the latency introduced by the iterative process renders the beamforming solution outdated.

Thanks to the recent advances in deep learning (DL) techniques, it becomes possible to learn the optimal beamforming in real time by taking into account both the performance and the computational delay simultaneously. This is because the DL technique trains neural networks offline and then deploys the trained neural networks for online optimization. The computational complexity is transferred from the online optimization to the offline training, and only simple linear and nonlinear operations are needed when the trained neural network is used to find the optimal beamforming solution, thus greatly reducing the computational complexity and delay.

However, with the exception of [7–10], there are no works focusing on the beamforming design in multi-antenna communications based on DL. [7] considered an outage-based approach to transmit beamforming in order to deal with the channel uncertainty at base stations (BSs). But only a single user was considered in [7]. [8] designed a decentralized robust precoding scheme based on DNN in a network MIMO configuration. The projection over a finite dimensional subspace in [8] reduced the difficulty, but also limited the performance. [9] used a DL model to predict the beamforming matrix directly from the signals received at the distributed BSs based on omni or quasi-omni beam patterns in millimeter wave systems, whose sum rate performance was restricted by the quantized codebook constraint. [8, 9] predicted the beamforming matrix in the finite solution space at the cost of performance loss. Furthermore, [7, 10] directly estimated the beamforming matrix without exploiting the problem structure in which the number of variables to predict increases significantly as the numbers of transmit antennas and users increase. This will lead to high training complexity of the neural networks when the numbers of transmit antennas and users are large.

Motivated by these facts, we propose a beamforming neural network (BNN) to deal with the power minimization problem. The proposed BNN is based on the CNN structure and the exploitation of expert knowledge i.e., uplink-downlink duality. The real and imaginary parts of complex channel coefficients are fed into the BNN as two vectors. Due to the parameter sharing scheme used in the CNN structure, less parameters are required. In the proposed BNN for the power minimization problem, instead of estimating the beamforming matrix with NK elements, where N is the number of the transmit antennas

†Jiangsu Key Laboratory of Wireless Communications, Nanjing University of Posts and Telecommunications, Nanjing 210003, P. R. China, e-mail: {2015010203, zhangjun}@njupt.edu.cn

¶Engineering Research Center of Health Service System Based on Ubiquitous Wireless Networks, Ministry of Education, Nanjing University of Posts and Telecommunications, Nanjing 210003, P. R. China.

§Wolfson School of Mechanical Electrical and Manufacturing Engineering, Loughborough University, Leicestershire LE11 3TU, U.K., e-mail: {g.zheng, y.zhu4}@lboro.ac.uk

‡School of Engineering and Digital Arts, University of Kent, Kent, CT2 7NT, UK, e-mail: j.z.wang@kent.ac.uk

£Department of Electrical & Computer Engineering Rutgers, State University of New Jersey, Piscataway, NJ 08854, e-mail: athinap@rutgers.edu

The work of W. Xia and J. Zhang was supported in part by the National Natural Science Foundation of China under Grant U1805262, Grant 61671251, and Grant 61871446, and in part by Nanjing University of Posts and Telecommunications Start-up Foundation under Grant NY215004. The work of G. Zheng was supported in part by the UK EPSRC under grant number EP/N007840/1, and the Leverhulme Trust Research Project Grant under grant number RPG-2017-129.

978-1-7281-2373-8/19/$31.00 ©2019 IEEE

at the BS and K is the number of users, we exploit the uplink-downlink duality of the solutions [5, 11] and predict the virtual uplink power allocation vector with only K elements. Since the optimal solutions to the power minimization problem exist, the proposed BNN updates network parameters using the supervised learning method which adopts the mean-squared error (MSE) as the metric in the loss function. Compared to the algorithm in [5], the proposed BNN solution can reduce the execution time per sample by two orders of magnitude, because it does not rely on an iterative process.

II. System Model

We consider a downlink transmission scenario where a BS equipped with N antennas serves K single-antenna users. The channel between user k and the BS is denoted as $\mathbf{h}_k = \sqrt{d_k}\tilde{\mathbf{h}}_k \in \mathbb{C}^{N\times 1}$ where $\tilde{\mathbf{h}}_k \sim \mathcal{CN}(\mathbf{0}, \mathbf{I}_N)$ and d_k are the small-scale fading and the large-scale fading, respectively. The received signal at user k is given by

$$y_k = \mathbf{h}_k^H \sum_{k'=1}^{K} \mathbf{w}_{k'} x_{k'} + n_k, \tag{1}$$

where $\mathbf{w}_k$ represents the beamforming vector for user k, $x_k \sim \mathcal{CN}(0, 1)$ is the transmitted symbol from the BS to user k, and $n_k \sim \mathcal{CN}(0, \sigma^2)$ denotes the additive Gaussian white noise (AWGN) with zero mean and variance σ^2. The received SINR of user k equals

$$\gamma_k^{dl} = \frac{|\mathbf{h}_k^H \mathbf{w}_k|^2}{\sum_{k'=1,k'\neq k}^{K} |\mathbf{h}_k^H \mathbf{w}_{k'}|^2 + \sigma^2}. \tag{2}$$

One conventional optimization problem seeks to minimize the power consumption under a set of SINR constraints, which has been investigated in many works [1–5, 12]. The power minimization problem can be formulated as

$$\begin{aligned} \min_{\mathbf{W}} \quad & \sum_{k=1}^{K} ||\mathbf{w}_k||^2 \\ \text{s.t. } & \gamma_k^{dl} \geq \Gamma_k, \forall k, \end{aligned} \tag{3}$$

where Γ_k is the SINR constraint of user k. For expressional ease, we define $\mathbf{\Gamma} = [\Gamma_1, \cdots, \Gamma_K]^T$ as the SINR constraint vector.

As mentioned in Section I, existing algorithms [4, 5] and tools, such as CVX, depend on iterative processes, which makes the beamforming technique unable to meet the demands of real-time applications. To address this challenge, DL is introduced since DL trains the neural network offline and finds solutions online only with some simple linear and nonlinear operations.

III. BNN for Power Minimization Problem

DL-based neural networks were initially designed for solving classification problems, but they can also achieve satisfactory performance in regression problems. For example, the DNN was used to predict transmit power [13, 14]. Existing works mainly take real data, such as channel gains and transmit power, as input and output, but channel and beamforming matrices are both complex. In addition, predicting the beamforming matrix with NK elements directly may lead to inaccurate and even under-fitting results. Obviously we can use wider or deeper neural networks with more neurons to improve the learning ability, but such a huge network will lead to high training and implementation complexities and cannot guarantee the learning performance. For example, too deep or wide neural networks can cause over-fitting. To overcome this challenge, instead of estimating the beamforming matrix directly, we introduce a scheme which first predicts the power allocation vector and then achieves the corresponding beamforming matrix based on the predicted results. In what follows we propose a BNN for the power minimization problem.

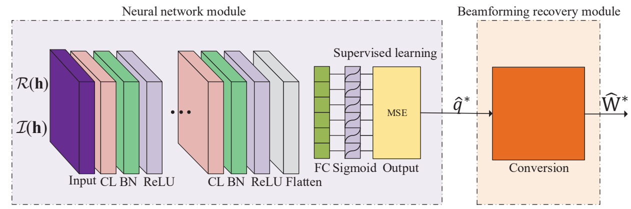


Fig. 1. BNN for the power minimization problem.

A. Uplink-Downlink Duality

Before we present the BNN for the power minimization problem, we first introduce the following lemma to describe the uplink-downlink duality of the power minimization problem [15].

Lemma 1. *Given the optimal beamforming matrix $\tilde{\mathbf{W}}^* = [\tilde{\mathbf{w}}_1^*, \ldots, \tilde{\mathbf{w}}_K^*]$ for the uplink problem, i.e.,*

$$\begin{aligned} \min_{\mathbf{q},\tilde{\mathbf{W}}} \quad & \sum_{k=1}^{K} q_k \\ \text{s.t. } & \gamma_k^{ul}(\tilde{\mathbf{W}}, \mathbf{q}) \geq \Gamma_k, \\ & ||\tilde{\mathbf{w}}_k||_2 = 1, \forall k, \end{aligned} \tag{4}$$

where $\gamma_k^{ul}(\tilde{\mathbf{W}}, \mathbf{q})$ is given as

$$\gamma_k^{ul}(\tilde{\mathbf{W}}, \mathbf{q}) = \frac{q_k |\mathbf{h}_k^H \tilde{\mathbf{w}}_k|^2}{\sum_{k'=1,k'\neq k}^{K} q_{k'} |\mathbf{h}_{k'}^H \tilde{\mathbf{w}}_k|^2 + \sigma^2}. \tag{5}$$

The optimal beamforming vectors $\mathbf{w}_k^*, \forall k$, for the downlink problem (3)*, can be obtained by multiplying the optimal normalized beamforming vector $\tilde{\mathbf{w}}_k^*$ by a scaling factor, i.e., $\mathbf{w}_k^* = p_k^* \tilde{\mathbf{w}}_k^*, \forall k$, where p_k^* is the k-th element of vector $\mathbf{p}^* = [p_1^*, \ldots, p_K^*]^T \in \mathbb{R}^{K\times 1}$ and*

$$\mathbf{p}^* = \sigma^2 \mathbf{\Psi}^{-1} \mathbf{1}, \tag{6}$$

where

$$[\mathbf{\Psi}]_{kk'} = \begin{cases} \frac{1}{\Gamma_k} |\mathbf{h}_k^H \tilde{\mathbf{w}}_k^*|^2, & \text{if } k = k', \\ -|\mathbf{h}_k^H \tilde{\mathbf{w}}_{k'}^*|^2, & \text{else.} \end{cases} \tag{7}$$

The vector of the scaling factors $\mathbf{p}^*$ is the optimal downlink power allocation vector. Given the optimal normalized beamforming matrix $\tilde{\mathbf{W}}^*$, **Lemma 1** allows us to achieve the optimal downlink power vector $\mathbf{p}^*$ by (6), then $\mathbf{W}^* = \tilde{\mathbf{W}}^*\mathbf{P}^*$ with $\mathbf{P}^* = \text{diag}(\mathbf{p}^*)$. Actually, if we know the uplink power allocation vector $\mathbf{q}$, the normalized beamforming matrix $\tilde{\mathbf{W}}$ can be inferred as

$$\tilde{\mathbf{w}}_k = \frac{\mathbf{T}^{-1}\mathbf{h}_k}{||\mathbf{T}^{-1}\mathbf{h}_k||_2}, \forall k, \tag{8}$$

where $\mathbf{T} = \sigma^2\mathbf{I}_N + \sum_{k=1}^K q_k\mathbf{h}_k\mathbf{h}_k^H$. Therefore, the only results that need to be predicted by the BNN is the uplink power allocation vector $\mathbf{q}$, which reduces significantly the computational complexity compared to the strategy that attempts to predict the beamforming matrix directly. The iterative algorithm, as shown in **Algorithm 1** [5], provides a way to achieve the optimal uplink power allocation vector as the training samples in the supervised learning method.

Algorithm 1 Iterative algorithm for power minimization problem [5].

Initialization: $\mathbf{q}^{(0)}$, $f^{(0)}$, and $i = 1$.
while $|f^{(i)} - f^{(i-1)}| > \epsilon$ **do**
  for $k = 1, 2, \ldots, K$ **do**
    $\mathbf{T}^{(i)} = \sigma^2\mathbf{I}_N + \sum_{k=1}^K q_k^{(i)}\mathbf{h}_k\mathbf{h}_k^H$.
    $q_k^{(i)} = \frac{\Gamma_k}{1+\Gamma_k}(\mathbf{h}_k^H(\mathbf{T}^{(i)})^{-1}\mathbf{h}_k)^{-1}$.
  end for
  $f^{(i)} = \sum_{k=1}^K q_k^{(i)}$.
  $i = i + 1$.
end while
$\tilde{\mathbf{w}}_k^{(i)} = (\mathbf{T}^{(i)})^{-1}\mathbf{h}_k, \forall k$.
$\tilde{\mathbf{w}}_k^{(i)} = \tilde{\mathbf{w}}_k^{(i)}/||\tilde{\mathbf{w}}_k^{(i)}||_2, \forall k$.
Output $\mathbf{q}^{(i)}$ and $\tilde{\mathbf{W}}^{(i)}$ as the optimal results $\mathbf{q}^*$ and $\mathbf{W}^*$.

B. BNN Structure

The proposed BNN for the power minimization problem, as shown in Fig. 1, is based on the CNN architecture. The reason for choosing the CNN, instead of other neural networks, is that the CNN can share parameters among the image and real parts of complex channel coefficients which are fed into the BNN, thus reducing the number of parameters. The proposed BNN includes two main modules: the neural network module and beamforming recovery module. The neural network module is composed of an input layer, convolutional (CL) layers, batch normalization (BN) layers, activation (AC) layers, a flatten layer, a fully-connected (FC) layer, and an output layer, whereas the beamforming recovery module includes a conversion layer, which maps/converts the key feature $\hat{\mathbf{q}}^*$ to the beamforming matrix. Note that the functional layers in the beamforming recovery module are designed according to the expert knowledge of the beamforming optimization. The expert knowledge is problem-dependent and has no unified form, but what is in common is that the expert knowledge can significantly reduce the number of variables to be predicted compared to the beamforming matrix. Below we give a brief introduction to these layers.

1) Input layer: The complex channel coefficients are fed into the neural network module to predict the key feature $\hat{\mathbf{q}}^*$, which are not supported by the current neural network software. To deal with this issue, two data transformations are available. One is to separate the complex channel vector, for example $\mathbf{h} = [\mathbf{h}_1^T, \cdots, \mathbf{h}_K^T]^T \in \mathbb{C}^{NK\times 1}$, into the in-phase component $\mathfrak{R}(\mathbf{h})$ and quadrature component $\mathfrak{I}(\mathbf{h})$, where $\mathfrak{R}(\mathbf{h})$ and $\mathfrak{I}(\mathbf{h})$ contain the real and imaginary parts of each element in $\mathbf{h}$, respectively. We call this transformation **I/Q transformation**. Another transformation, suggested by [16], is to map the complex channel vector $\mathbf{h}$ into two real vectors $\mathfrak{P}(\mathbf{h}_k)$ and $\mathfrak{M}(\mathbf{h}_k)$, where the former contains the phase information and the latter includes the magnitude information of $\mathbf{h}$. This transformation is referred to as **P/M transformation**. Without loss of generality, we adopt **I/Q transformation** of complex channels as the input of the first CL layer. Each CL layer creates one or more convolution kernels that are convolved with the layer input and the parameters of convolution kernels are shared among different channel coefficients. Note that the samples are fed into the neural network module in batches.

2) BN Layer: The BN layers are introduced in the neural network module, which can be put before or after the AC layers [17] according to practical experience. In the proposed framework, we adopt the former where the BN layers normalize the output of the CL layers through subtracting the batch mean and dividing by the batch standard deviation. Consequently, the BN operation introduces two trainable parameters, i.e., a “mean” parameter and a “standard deviation” parameter, in each BN layer. The denormalization is allowed by changing only the two parameters, instead of changing all parameters which may lead to the instability of the neural network module. Besides, the BN layer has the following advantages:

- The probability of over-fitting is reduced since the BN layer presents some regularization effects similar to dropout, by adding some noise to each AC layer.
- The BN layer enables a higher learning rate which can accelerate convergence because the BN operation can avoid the AC function going into the gradient-insensitive region.
- In addition, with the BN layer, the neural network is less sensitive to the initialization of weights.

3) Activation Layer: Since the predicted variables are continuous and positive real numbers, it is suggested that the AC functions that can generate negative values, such as tanh and linear functions should not be used in the last AC layer. The rectified linear unit (ReLU) and sigmoid functions are good choices for the last AC layer. For the intermediate AC layers, the ReLU function generally shows better performance than other AC functions. However, if the BN layer is adopted before each AC layer, the sigmoid function can also work well because of the normalization operation introduced by the BN layer.

4) Flatten Layer and Output Layer: The flatten layer is only used to change the shape of its input into the correct format, i.e., a vector, for the FC layer to interpret. The main function of the output layer is to generate the predicted results after the neural network finishes training.

Note that apart from these functional layers, the loss function also plays an important role in the proposed BNN, which is marked on the output layer in Fig. 1. The loss function together with the learning rate guides the learning process of the neural network. In other words, the loss function "tells" the neural network how to update its parameters. Since the output values are continuous, it is suggested to utilize the mean absolute error (MAE) or the MSE as a metric.

5) Conversion Layer: After receiving the scaled power allocation vector $\hat{\mathbf{q}}^*$, we can achieve the downlink beamforming matrix $\hat{\mathbf{W}}^*$ as the final output of the BNN based on $\hat{\mathbf{q}}^*$ by the conversion layer. The beamforming recovery implemented by the conversion layer includes the following process:

1) Calculate $\mathbf{T}^* = \sigma^2 \mathbf{I}_N + \sum_{k=1}^{K} \hat{q}_k^* \mathbf{h}_k \mathbf{h}_k^H$.
2) Calculate $\tilde{\mathbf{w}}_k^* = \tilde{\mathbf{w}}_k^* / ||\tilde{\mathbf{w}}_k^*||_2, \forall k$, where $\tilde{\mathbf{w}}_k^* = (\mathbf{T}^*)^{-1}\mathbf{h}_k$.
3) Calculate the downlink power allocation vector $\hat{\mathbf{p}}^* = \sigma^2(\mathbf{\Psi}^*(\tilde{\mathbf{W}}^*, \mathbf{\Gamma}))^{-1}\mathbf{1}$.
4) Output the downlink beamforming vectors $\hat{\mathbf{w}}_k^* = \hat{p}_k^* \tilde{\mathbf{w}}_k^*, \forall k$, as the final results.

Note that the predicted power vector $\hat{\mathbf{q}}^*$ by the BNN is, in general, not exact. The prediction error will lead to the inaccuracy of power allocation vector $\hat{\mathbf{p}}^*$ as well as the downlink beamforming $\hat{\mathbf{W}}^*$. More specifically, if the predicted power vector $\hat{\mathbf{q}}^*$ has an acceptable accuracy with respect to the target power vector $\mathbf{q}^*$, i.e., $||\mathbf{q}^* - \hat{\mathbf{q}}^*||_2^2 < \varepsilon$ where ε is a small constant, then we can obtain a suboptimal solution whose objective value is larger than that of the optimal solution, i.e., $\sum_{k=1}^{K} ||\hat{\mathbf{w}}_k^*||_2^2 > \sum_{k=1}^{K} ||\mathbf{w}_k^*||_2^2$. Intuitively, The extra power consumption $q_{extra} = \sum_{k=1}^{K} ||\hat{\mathbf{w}}_k^*||_2^2 - \sum_{k=1}^{K} ||\mathbf{w}_k^*||_2^2$ can be regarded as the cost of the prediction error. However, if the predicted vector $\hat{\mathbf{q}}^*$ has a significant error, i.e., $||\mathbf{q}^* - \hat{\mathbf{q}}^*||_2^2 \gg \varepsilon$, the downlink beamforming $\hat{\mathbf{W}}^*$ inferred from the prediction $\hat{\mathbf{q}}^*$ may become infeasible since some elements of the vector $\hat{\mathbf{p}}^*$ have negative values. This suggests that there is a certain probability of infeasibility of the BNN prediction for the power minimization problem. However, our experiments show that the failure probability of the proposed BNN for the power minimization problem is lower than 1% in most settings. More details will be given in Section IV.

IV. Simulation Results

To evaluate the performance of the proposed BNN, we carry out numerical simulations to compare the BNN solution with two benchmark solutions, i.e., zero-forcing (ZF) beamforming and the optimal solution obtained using **Algorithm 1**. We consider a downlink transmission scenario where the BS is equipped with $N = 6$ antennas and its coverage is a disc with a radius of 500 m. There are $K = 4$ single-antenna users and these users are distributed uniformly within the coverage of the BS. Note that none of these users is closer to the BS than 100 m. The pathloss between the user and the BS is set as $128.1 + 37.6 \log_{10}(\omega)$[dB] [18] where ω is the distance in km.

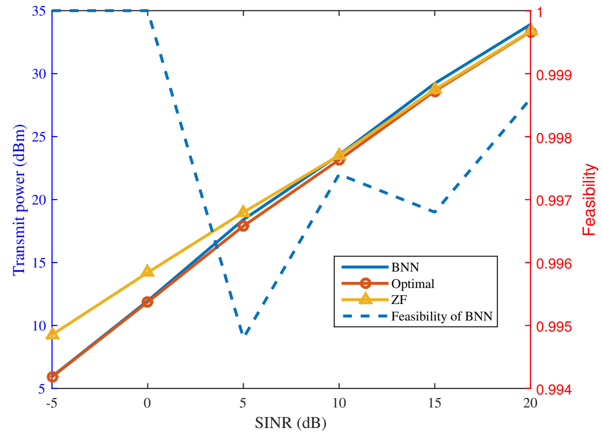


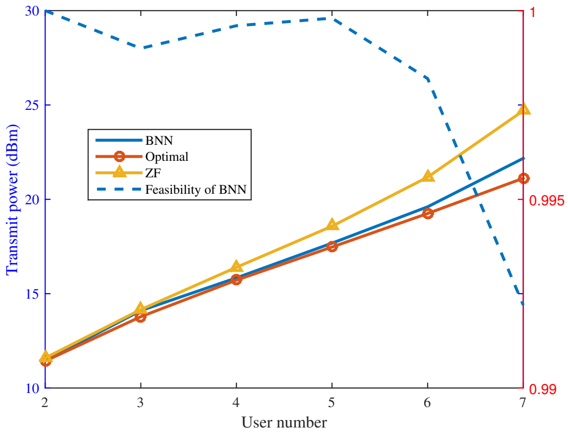
Fig. 2. The power performances averaged over 5000 samples under $\{K = 4, N = 6\}$.

The noise power spectral density is $\sigma^2 = -174$ dBm/Hz and the total system bandwidth is 20 MHz. Besides, perfect CSI is assumed to be available at the BS.

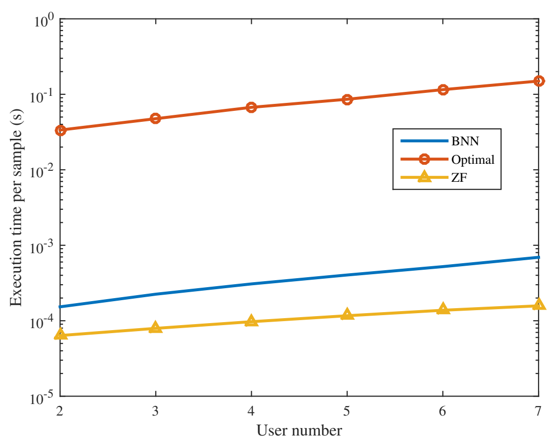
In our simulation, we prepare 20000 training samples and 5000 testing samples, respectively. The proposed BNN has one input layer, two BN layers, two CL layers, three AC layers, one flatten layer, one FC layer, and one output layer. The FC layer has K neurons and each CL layer has 8 kernels of size 3×3. Furthermore, Adam optimizer is used with the MSE metric-based loss function and the first two AC layers adopt the ReLU function. Note that the last AC layer can be the ReLU or sigmoid function. Here, we adopt the sigmoid function so that the target output in the training and testing samples should be normalized. The proposed BNN solutions are implemented in Python 3.6.5 with Tensorflow 1.2.1 and Keras 2.2.2 on a computer with 1 Intel i7-7700U CPU Core and RAM of 32GB.

In Fig. 2, we compare the power performances of the optimal beamforming, ZF beamforming, and BNN solution, under the assumption that the SINR constraints of all users are the same, i.e. $\Gamma_k = \Gamma, \forall k$. Note that Fig. 2 has two Y-axes where the left Y-axis is used to measure the transmit power and the right Y-axis is used to describe the feasibility of the BNN. As mentioned in Section III, the BNN may fail to find a feasible solution to problem **P2** if the prediction error is unacceptable. Fig. 2 demonstrates that the power performance of the BNN solution is close to that of the optimal solution, and significantly outperforms the ZF beamforming in the low SINR-constraint regime which is higher than that of the optimal solution. Besides, we find that the feasibility of the BNN solution is more than 99.4%.

To further compare the BNN solution with the optimal solution and the ZF beamforming, we plot their power performance and execution time per sample in Figs. 3(a) and 3(b), respectively. In Fig. 3, the BS antenna number and SINR target of users are fixed as $N = 8$ and $\Gamma = 5$ dB. It is observed from Fig. 3(a) that as the user number K increases,



(a)



(b)

Fig. 3. Comparison of three different beamforming solutions, i.e., the optimal solution, BNN solution, and ZF beamforming: (a) power performance and (b) execution time per sample averaged over 5000 samples under $\{\Gamma = 5$ dB, $N = 8\}$.

the performance gap between the ZF beamforming and the optimal beamforming becomes large because more users share the array gain. The BNN solution shows a better performance than ZF beamforming and has the feasibility of at least 99%. Fig. 3(b) demonstrates that compared to the optimal solution, the BNN solution can reduce the execution time per sample by two orders of magnitude, which is slightly longer than that of the ZF beamforming. This is because the BNN solution and the ZF beamforming are obtained without an iterative process, but the BNN needs to execute the neural network operations as well as the conversion process. According to the results in Figs. 3(a) and 3(b), we can conclude that the BNN solution provides a good balance between the performance and the computational complexity.

V. Conclusion

In this work, we proposed a BNN based on the CNN structure for the power minimization problem under a set of QoS constraints. The supervised learning method was adopted for the power minimization problem because its optimal solutions exist for generating training samples. Furthermore, in order to reduce the complexity of prediction, the proposed BNN takes advantage of expert knowledge to extract the key features instead of predicting the beamforming matrix directly. Simulation results demonstrated that the proposed BNN solutions provided a good balance between the performance and computational complexity.

References

[1] A. B. Gershman, N. D. Sidiropoulos, S. Shahbazpanahi, M. Bengtsson, and B. Ottersten, "Convex optimization-based beamforming," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 62–75, May 2010.

[2] A. Wiesel, Y. C. Eldar, and S. Shamai, "Linear precoding via conic optimization for fixed MIMO receivers," *IEEE Trans. Signal Process.*, vol. 54, no. 1, pp. 161–176, Jan. 2006.

[3] Z.-Q. Luo, W.-K. Ma, A. M.-C. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, May 2010.

[4] F. Rashid-Farrokhi, L. Tassiulas, and K. R. Liu, "Joint optimal power control and beamforming in wireless networks using antenna arrays," *IEEE Trans. Commun.*, vol. 46, no. 10, pp. 1313–1324, Oct. 1998.

[5] Q. Shi, M. Razaviyayn, M. Hong, and Z. Luo, "SINR constrained beamforming for a MIMO multi-user downlink system: Algorithms and convergence analysis," *IEEE Trans. Signal Process.*, vol. 64, no. 11, pp. 2920–2933, Jun. 2016.

[6] H. Zhu, "Radio resource allocation for OFDMA systems in high speed environments," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 4, pp. 748–759, May 2012.

[7] Y. Shi, A. Konar, N. D. Sidiropoulos, X. Mao, and Y. Liu, "Learning to beamform for minimum outage," *IEEE Trans. Signal Process.*, vol. 66, no. 19, pp. 5180–5193, Oct. 2018.

[8] P. de Kerret and D. Gesbert, "Robust decentralized joint precoding using team deep neural network," in *Proc. Int. Symp. Wireless Commun. Systems (ISWCS)*, Lisbon, Portugal, Aug. 2018.

[9] A. Alkhateeb, S. Alex, P. Varkey, Y. Li, Q. Qu, and D. Tujkovic, "Deep learning coordinated beamforming for highly-mobile millimeter wave systems," *IEEE Access*, vol. 6, pp. 37 328–37 348, 2018.

[10] H. Huang, W. Xia, J. Xiong, J. Yang, G. Zheng, and X. Zhu, "Unsupervised learning based fast beamforming design for downlink MIMO," *IEEE Access*, pp. 1–1, 2018.

[11] M. Schubert and H. Boche, "Solution of the multiuser downlink beamforming problem with individual SINR constraints," *IEEE Trans. Veh. Technol.*, vol. 53, no. 1, pp. 18–28, Jan. 2004.

[12] F. Rashid-Farrokhi, K. R. Liu, and L. Tassiulas, "Transmit beamforming and power control for cellular wireless systems," *IEEE J. Sel. Areas Commun.*, vol. 16, no. 8, pp. 1437–1450, Oct. 1998.

[13] F. Liang, C. Shen, W. Yu, and F. Wu, "Towards optimal power control via ensembling deep neural networks," *arXiv preprint arXiv:1807.10025*, 2018.

[14] H. Sun, X. Chen, Q. Shi, M. Hong, X. Fu, and N. D. Sidiropoulos, "Learning to optimize: Training deep neural networks for wireless resource management," in *Proc. IEEE Int. Workshop Signal Process. Advances Wireless Commun. (SPAWC)*, Sapporo, Japan, Jul. 2017, pp. 1–6.

[15] W. Yu and T. Lan, "Transmitter optimization for the multi-antenna downlink with per-antenna power constraints," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2646–2660, Jun. 2007.

[16] M. Kulin, T. Kazaz, I. Moerman, and E. D. Poorter, "End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications," *IEEE Access*, vol. 6, pp. 18 484–18 501, 2018.

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[18] H. Dahrouj and W. Yu, "Coordinated beamforming for the multicell multi-antenna wireless system," *IEEE Trans. Wireless Commun.*, vol. 9, no. 5, pp. 1748–1759, May 2010.