Deep Learning Based Antenna Muting and Beamforming Optimization in Distributed Massive MIMO Systems

Yu Chen[1,2](✉), Kai Zhao[1], Jing-ya Zhao[1], Qing-hua Zhu[1], and Yong Liu[1]

[1] School of Telecommunication Engineering, Beijing Polytechnic, Beijing 100176, People's Republic of China
buptchen@163.com
[2] School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, People's Republic of China

Abstract. Inspired by the success of Deep Learning (DL) in solving complex control problems, a new DL-based approximation framework to solve the problems of antenna muting and beamforming optimization in distributed massive MIMO was proposed. The main purpose is to obtain a non-linear mapping from the raw observations of networks to the optimal antenna muting and beamforming pattern, using Deep Neural Network (DNN). Firstly, the antenna muting and beamforming optimization problem is modeled as a non-combination optimization problem, which is NP-hard. Then a DNN based framework is proposed to obtain the optimal solution to this complex optimization problem with low-complexity. Finally, the performance of the DNN-based framework is evaluated in detail. Simulation results show that the proposed DNN framework can achieve a fairly accurate approximation. Moreover, compared with the traditional algorithm, DNN can be reduced the computation time by several orders of magnitude.

Keywords: Deep learning · Distributed massive MIMO · Deep Neural Network · Antenna muting · Beamforming

1 Introduction

In recent years, with the increasing number of smart devices, various wireless services have been exploding, such as social networks, Internet of Things, high-quality radio video streams, etc. The resulting wireless data flow has also been increased by dozens of times. This has led to an increase in the cost of building, operating, and upgrading radio access network, while revenues have grown slowly. As a result, the massive multiple input multiple output (MIMO) technique [1], is proposed to handle the ever-increasing mobile traffic demands.

Characterized by the large number of antennas at the front-end of the radio frequency (RF) chains and the centralized signal processing, massive MIMO not only significantly improves the spectral efficiency, but also greatly reduces the energy

V. C. M. Leung et al. (Eds.): 5GWN 2019, LNICST 278, pp. 18–30, 2019.
https://doi.org/10.1007/978-3-030-17513-9_2

consumption. Hence the massive MIMO is deemed as one of the key techniques in the next generation mobile networks, i.e., 5G. The massive antennas are originally deployed on a single array in a centralized manner, which can achieve the channel harden and favorable propagation. However, recent studies [2] find that distributing the massive antennas not only inherent the advantage of centralized massive MIMO in terms of channel harden and favorable propagation, but also achieve the huge diversity gains through cooperative beamforming, which leads to superior capacity performance.

On the other hand, mobile traffic varies in the spatial and time domain. Always keeping all antennas of the distributed massive MIMO systems active is energy inefficient in the low traffic scenario. The large number of antennas would result in considerable energy consumption while contribute little to the QoS guarantee of users in such cases Hence, adaptively muting the transmission of some antennas according to the dynamic traffic demands seems to be a promising scheme to improve the energy efficiency.

Dynamically turning off the hardware of wireless networks, such as antennas, RF chains at the base stations (BSs), is a commonly used method to reduce the energy consumption, as can be seen in a survey [4]. Zhou et al. [5] proposed a traffic aware BS sleeping control scheme, which can achieve great energy efficiency gain. Alberto et al. [6] proposed cell wilting and blooming scheme to enable active BSs dynamically adjust the cell size to serve the users originally located in the coverage of sleeping BSs. In addition, cooperative beamforming based on the distributed antennas is well recognized to improve the performance of hardware sleeping. Niu et al. [7] proposed to utilize the cooperative beamforming technique to compensate the coverage hole caused by the BS sleeping. Whereas Shi et al. [8] focus a distributed massive MIMO network and proposed a group sparsity based joint cooperative beamforming and antenna muting algorithm. The results show that joint optimization of antenna muting allows more antennas stay in inactive states and hence achieves higher energy efficiency.

However, the previous works all solve the joint optimization problem of antenna muting and cooperative beamforming from the point of numerical optimization. Numerical optimization has played a key role in solving the problem of wireless resource management. In the literature, many kinds of handcrafted algorithms are developed to find a stable solution for the antenna muting and cooperative beamforming for certain scenarios. Nevertheless, Considering the NP hardness of the problem, such algorithms has high computing complexity and only suitable for certain scenarios. In addition, they also cause a serious gap between theoretical analysis design and practical application.

Deep learning has shown great potential and advantages in feature extraction and model fitting [9], thereby attracting a large number of scholars to study its theory and application, and it has developed rapidly in academia and industry. Inspired by its superior performance in solving complex control problems [9], in this paper, we aim to design low complex deep learning solution to find the optimal antenna muting and cooperative beamforming pattern in the distributed massive MIMO systems.

2 System Model

In this section, a typical distributed massive MIMO is considered. Then we formulate the signal model and power consumption model considering the antenna muting and cooperative beamforming.

2.1 Distributed Massive MIMO System

Consider a single cell of distributed massive MIMO system, as shown in Fig. 1. The single cell consisting of a large number of distributed antennas which jointly serve multiple users in the form of cooperative beamforming. Let $\mathrm{R} : R = \{1, 2, \ldots, \mathrm{R}\}$ denote the set of antennas and $\mathrm{U} : U = \{1, 2, \ldots, \mathrm{U}\}$ denote the set of user equipments (UEs). Each UE is assumed to be equipped with a single antenna. All the distributed antennas are connected to a centralized signal processing unit (denoted as the CPU), which is a cloudified processing pool. The CPU is able to collect and share the channel state information and user's requested data among the antennas in a centralized manner.

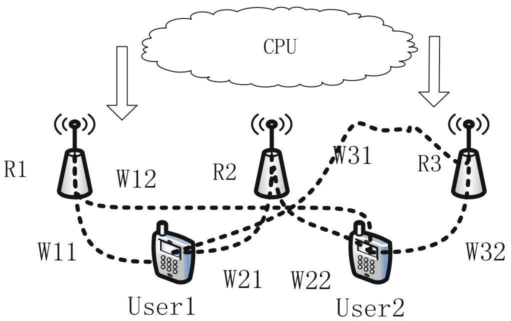


Fig. 1. The illustration of distributed massive MIMO

The channel between UEs and antennas are assumed to be block flat, i.e., the channel gain is constant during a certain time slot. At each time slot, the CPU is responsible for determining the antenna muting and cooperative beamforming pattern. The decision made is then distributed to the antennas. Upon received the muting and cooperative beamforming decision from the CPU, each antenna changes its state, namely keeping transmitting the data stream or enter the muting state. The transmitted symbols of UEs are assumed to be a Gaussian random variable with zero mean and unit variance. UEs receive the required data symbols from all the active antennas which performs cooperative beamforming. The corresponding signal-to-interference-plus-noise ratio (SINR) of each UE can be expressed as

$$SINR_u = \frac{\left|h_u^H w_u\right|^2}{\sum_{v \neq u} \left|h_u^H w_v\right|^2 + \sigma^2}, u \in U \tag{1}$$

where $h_u := [h_{1u}, h_{2u}, \ldots, h_{Ru}]^T$ represents the channel gain vector, and each element h_{ru} represents the channel gain from antenna r to UE u; similarity, the cooperative beamforming weights are stacked in a vector $w_u = [w_{1u}, w_{2u}, \ldots, w_{Ru}]^T$, wherein each element w_{ru} represents the beamforming weight from antenna r to UE u. Generally, the channel coefficients and the beamforming weights are all complex values, but in this study, in order to facilitate the algorithm, the modulus values of complex numbers were taken to work in the real-value domain, σ^2 is background noise during transmission.

By using the Shannon Hartley theorem, the maximum data transmission rate that each user u can achieve is given by:

$$R_u = B\log_2\left(1 + \frac{SINR_u}{\Gamma_m}\right),\ u \in U \tag{2}$$

Where, B is the channel bandwidth; Γ_m is SNR (Signal-to-Noise Ratio) interval, which depends on the specific modulation scheme.

When specifically considering a distributed massive MIMO network, the power consumption depends on the functionality splitting between the CPU and antennas. Herewith, we adopt a simple power consumption model similar to [10–12], which characterize the impact of the muting antenna on the total power consumption. This model is applicable to different types of base stations, i.e., there's basically a linear function relationship between the transmit power and the power consumption of the cell, and such a relationship is also applicable to the relationship between transmit power and the power consumption of the antennas. Therefore, an empirical linear functional relationship can be adopted for modeling each antenna:

$$P_r = \begin{cases} P_{r\in A,active} + \frac{1}{\eta_l}P_{r,trans} \\ P_{r\in S,sleep} \end{cases} \tag{3}$$

Where, η_l is a constant, indicating the drain efficiency of the power amplifier, and it is a constant. $P_{r,active}$ is the power consumption of an antenna in active state, which is power consumption necessary for the basic operation of maintaining the antenna in the transmission working state. If an antenna does not have any information that can be transmitted and is not selected to be active for transmitting signals, then it will be in muting mode. In this mode, the power consumption of the antenna is $P_{r,sleep}$, in order to save energy, there's the need to transfer antenna into sleep mode as much as possible, thus to reduce the antenna in active state as much as possible. Antenna has two states, active and sleep, $A \subseteq R$ and $S \subseteq R$ represents respectively active set and sleep set of the antennas, and $A \cup S = R$. $P_{r,trans}$ is the transmit power of antenna and it satisfies the following relationship:

$$P_{r,trans} = \sum\nolimits_{r\in A}\sum\nolimits_{u\in U}\left|w_{r,u}\right|^2 \tag{4}$$

Based on above analysis, the total transmission power consumption of the entire system in the current period can be expressed as:

$$P(A,S,G) = \sum_{r \in A} \sum_{u \in U} \frac{1}{\eta_l} \left| w_{r,u} \right|^2 + \sum_{r \in A} P_{r,active} + \sum_{r \in S} P_{r,sleep} \tag{5}$$

Where, the first term represents power consumption associated with the signal transmission process, while the second and third terms are the total power consumption of the antenna in active and sleep states respectively to maintain basic operation.

In order to minimize the total transmission power consumption, there are two strategies:

(1) Optimize the cooperative beamforming to reduce the transmit power;
(2) Reduce the number of active antennas and the corresponding transmission links.

However, the above two strategies are conflicting. In order to reduce the transmission power consumption, more antennas need to be activated to obtain the higher beamforming gain; but if more antennas are allowed to be activated, it will increase the power consumption of the transmission link. Therefore, in order to minimize network power consumption, joint optimization of the antenna (and corresponding transmission link) selection and transmit beamforming are required.

3 Problem Formulation

A very important part of transmission power model proposed in previous section is power consumption in the process of transmitting signals, which depends on the cooperative beamforming weights of each antenna. In order to minimize the total power consumption of the entire system based on meeting the user's needs, it's necessary to considering the joint antenna muting and cooperative beamforming design, and this problem can be expressed mathematically as follows:

$$\underset{A,\ \{w_{r,u}\}}{\text{minimize}} \sum_{r \in A} \sum_{u \in U} \left| w_{r,u} \right|^2 \tag{6}$$

In order to ensure that the needs of each user are met, the following constraints are required:

$$SINR_u \geq \gamma_u,\ u \in U \tag{7}$$

$$\gamma_u = \Gamma_m (2^{R_u/B} - 1) \tag{8}$$

Total transmit power sum of each antenna r is constrained by the following equation:

$$\sum_{u \in U} \left| w_{r,u} \right|^2 \leq P_r, r \in A \tag{9}$$

Where, R_u is the requirement of each user u, and P_r is the maximum allowed transmit power of all antennas. As can be seen, the optimization variables of the above combination optimization problem are the antenna muting decision A, and also the cooperative beamforming weights from each antenna to each UE $\{w_{r,u}\}$. The overall goal is to minimize total power consumption. It should be noted that the factor η_l in the power model (5) and other related parameters are ignored here because they are just constants. Herewith, the inequality (7) can also be rewritten as:

$$\sum_{v \in U} \left| h_u^H w_v \right|^2 + \sigma^2 \leq \frac{1+\gamma_u}{\gamma_u} \left| h_u^H w_u \right|^2 \tag{10}$$

Given the need R_u of each user u, γ_u can be calculated according to Eq. (10) accordingly. Note that, if the antenna muting pattern A is known, this optimization problem can be transformed into a second-order cone programming (SOCP) problem, which can be efficiently solved by the convex optimization tools such as the CVX toolbox in the MATLAB. However, here the antenna muting pattern A and the cooperative beamforming weights need to be simultaneously solved. This makes this problem NP-hard and difficult to solve.

4 DNN Based Joint Antenna Muting and Beamforming Optimization

In this section, a brief introduction of DNN and DNN approximation theory is first present. Then a DNN approximate framework is constructed in order to establish a non-linear mapping between the network states and the optimal antenna muting and cooperative beamforming solution.

4.1 DNN Approximation Theory

For any deterministic algorithm that uses iteration to represent continuous mapping, its initial value can be used as an additional input variable, and then the trained neural network can be used to learn the given algorithm behavior [13]. If the potential optimization problem is a nonconvex problem with the multiple solutions, then it is necessary to use the initial value as one of the input variables, because if there is no initial value, the mapping cannot be well defined, that is, it may converge to multiple isolated solutions. Or one can learn the mapping relation well by fixing the initial value of the solution.

It can be verified that the iteration in each of the convex optimization problem solving processes represents a continuous mapping, and the optimization variables are in a compact set [14]. Therefore, if the channel implementation set $\{h_{ij}\}$ is assumed to be in a compact set, the solution can be arbitrarily approximated by the feedforward network with multiple hidden layers.

4.2 Construction of DNN Optimization System

DNN Structure Setting

The method proposed in this study uses a fully connected neural network with an input layer, multiple hidden layers and an output layer. The specific structure is shown in Fig. 2. The input of the neural network is a vector $h_u := [h_{1u}, h_{2u}, \ldots, h_{Ru}]^T$ of the magnitude of the channel coefficients. There are three network outputs, which are respectively beamforming matrix $w_u = [w_{1u}, w_{2u}, \ldots, w_{Ru}]^T$, the antenna state (even if each antenna in the network is on or off), and the optimal transmission power. In addition, the ReLU function is used as the activation function of the hidden layer.

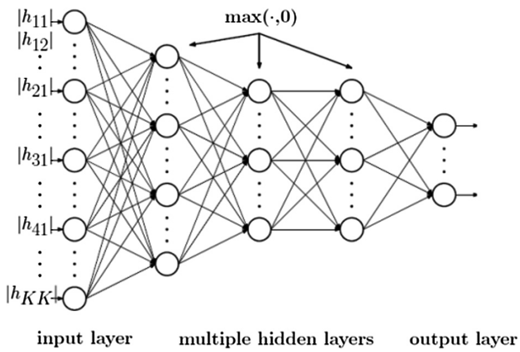


Fig. 2. Structure of DNN

Data Generation

The data is generated in the following manner. The modulus value $\{|h_{Ru}^{(i)}|\}$ of channel coefficients is first generated according to a particular distribution (described in more detail below), wherein superscript (i) is used to represent the index of training sample. For the sake of simplicity, the fixed maximum power $P_{\max}$ and ambient noise σ^2 for all the antennas were obtained. Then for each input tuple $(P_{\max}, \{|h_{Ru}^{(i)}|\}, \sigma)$, there is a corresponding optimal control result. In other words, the state set of the optimal node antenna, the beamforming matrix and the power allocation vector. Using $v_k^0 = \sqrt{P_{\max}}, \forall k$ as a fixed initial power value. Regarding the termination of the iteration, $obj_{new} - obj_{old} < 10^{-5}$ or iteration with times >500 can be chosen as the standard. The corresponding input-output relationship tuple $\left(\left\{\left|h_{Ru}^{(i)}\right|\right\}, \left\{\left|w_{Ru}^{(i)}\right|\right\}, P^{(i)}, A^{(i)} \subseteq R\right)$ is called as i^th training sample. Then is to repeat above process for multiple times, the entire training data set can be generated.

In the training phase of the neural network, the main function of verifying data set is to cross-validate the accuracy of generated network. Namely, randomly extracting a part from a set of measured data for modeling, and the rest of the data is used to verify the model to make the model selection. For the different models trained in the training set, their errors are counted, and the model with the smallest error is taken as the final model; and the network training should be terminated in time to avoid making the training depth too deep. In particular, the verification data set is relatively small compared to the amount of data in the training data set. In addition, T and V are respectively used to collect the index of the training set and data set.

Training Phase

In the training phase, the data set $\left(\left\{\left|h_{Ru}^{(i)}\right|\right\},\left\{w_{Ru}^{(i)}\right\},P^{(i)},A^{(i)}\subseteq R\right)_{i\in T}$ generated by the above process is first used to optimize the weight of the neural network. The loss function used is mean-square error between the modulus values of each element corresponding to the algorithm of solving the convex optimization problem and the beamforming matrix of the neural network output. The loss function (or cost function) is used to measure the extent of fit to a function. The smaller the value of the loss function, the higher the fit of the model is. The optimization algorithm used is an effective implementation of the small batch stochastic gradient descent algorithm, which is known as the RMSprop algorithm [10], which is an adaptive learning rate method that divides the gradient by the running square value of its nearest corresponding amplitude. By introducing an attenuation coefficient, the gradient cumulant is attenuated by a certain ratio per turn. Specifically, it first randomly extracts a set of samples with the capacity of m from the training data set and its corresponding output. Then is to calculate the gradient and error, and update the gradient cumulant r, and update other variables according to r and the calculation parameters of the gradient at this time. The specific update steps are shown in Table 1. The algorithm can automatically change the learning rate and alleviate the rapid change of the learning rate. That is, if the gradient is too large, the learning rate should be attenuated faster, and if the gradient is smaller, the learning rate should be slower. Moreover, if the depth is deep in the learning process, the algorithm does not have the problem of learning to end prematurely, which is very suitable for dealing with non-stationary targets.

Specifically, the layer-by-layer training method should be used to initialize network weights in the training process. First is to train a network with only one hidden layer. Only after this layer of network training is finished, the training of a network with two hidden layers can be started, and so forth [11]. Through this method, the problem of data acquisition can be solved effectively, thus to obtain a better local extremum. Regarding the network construction, it is necessary to train the first self-encoder with the original input data and learn the first-order feature of the original input. The first-order feature is used as the input of the self-encoder to learn and obtain the second-order feature; then the second-order feature is used as the input of the classifier, thus to obtain a model of mapping the second-order feature through training. Finally, these three layers are combined to construct a self-coded deep neural network with the two hidden layers [12].

In this study, the attenuation rate was chosen to be 0.9 according to the recommendations in literature [9]. In addition, the appropriate maximum training number (max epoch) and L2 regularization coefficient were chosen through the cross-validation of the training set and the verification set, the verification data was used for multiple times to continuously adjust parameters. In order to further improve the network performance generated by the training, the truncated normal distribution generation variable is used to initialize the network weight. Namely, first is to generate a variable according to the standard normal distribution, and then is to judge its absolute value. If it is greater than 2, it will be discarded and regenerate the new variable. In addition, the weight of each neuron is divided by the square root of its input number, which is to normalize the variance of each neuron output.

Test Phase

In the test phase, the performance of the trained neural network will be tested, and the channel parameters will be generated as the test data set according to the same distribution as the training phase. The function is to detect constructed neural network and effectively measure the deviation of the algorithm to evaluate the model accuracy. This method can test the performance of the neural network, and evaluate the performance of the results obtained based on the robustness of the model obtained through the training phase.

During the test, the first is to make the generated test data set pass through the trained neural network, and the optimized power distribution and beamforming results are also collected. Followed by is to compare the similarities between the resource allocation results generated by the DNN and the results of the numerical optimization algorithm. The higher the similarity, the more accurate the DNN is constructed.

Table 1. RMSProp algorithm

algorithm
Selecting global learning rate ε , attenuation rate ρ , and initial parameters θ
Selecting the small constant δ , usually setting to 10^{-6} , which is used for fixing the value when it is divided by the decimal
Initialize cumulative variable r=0
While does not reach the stop criterion **do**
Collecting a small batch $\left\{x^{(1)},\ldots,x^{(m)}\right\}$ containing m samples from the training set, and the corresponding target is $y^{(i)}$
Calculative gradient: $g \leftarrow \frac{1}{m}\nabla\theta\sum_i L(f(x^{(i)};\theta),y^{(i)})$
Cumulative squared gradient: $r \leftarrow \rho r+(1-\rho)g\odot g$
Calculation parameter update: $\Delta\theta=-\frac{\varepsilon}{\sqrt{\delta+r}}\odot g$ (element-by-element application $\frac{1}{\sqrt{\delta+r}}$)
Application update: $\theta \leftarrow \theta+\Delta\theta$
End while

5 Simulation and Performance Evaluation

5.1 Simulation Environment

Joint antenna muting and cooperative beamforming problem solving

This problem belongs to a class of non-convex optimization problems. Hence to find an optimal solution, we use the greedy scheme, where each combinations of antenna states (on/off) are explored, and for each given antenna muting pattern, we transformed the cooperative beamforming problem into a convex problem and solved it using mature toolbox.

DNN model construction

The model constructed in this study is a neural network with multiple hidden layers, which can be used to solve the problem of processing complex data. The method of training the multilayer neural network is to train one layer at a time, and a special network type called Auto encoder for each hidden layer can be trained. An automatic encoder is a layer in a neural network, and the goal is to copy its input at the output of the neural network. When the number of neurons in the hidden layer is less than the size of its input, the auto-encoder compresses it to express its input. The auto-encoder uses regularizer to learn the sparse representations in the first layer of the network, which can control the effects of the regularizer by setting various parameters. After building a complete DNN using the training data, the network is optimized by using the validation set data, which performs the backpropagation over the entire multi-layer network to improve network performance and minimize the loss function. In this study, the mean-square error of the modulus value of each element in the beamforming matrix was chosen as the loss function.

5.2 Channel Models

Each channel coefficient is generated according to a standard normal distribution, that is, a Rayleigh fading distribution with the zero mean and unit variance. Rayleigh fading distribution is a reasonable channel model and is widely used to simulate the performance of various resource allocation algorithms [13]. In the simulation of this study, five different network scene settings were considered, the numbers of antenna and user were respectively (32,8), (64,8), (128,16). In these different scene settings, the calculation will be conducted, and the similarity of the traditional optimization algorithm, DNN power consumption and beamforming matrix will be compared to evaluate its performance. So as to verify whether the input-output relationship learned by DNN can accurately approximate to the optimization algorithm, that is, to minimize the power consumption while satisfying user requirements.

5.3 DNN Parameter Selection

This section selected the following parameters for the deep neural network: maximum number of training options was five parameters from 50 to 1000, and the selection of L2 regularization coefficient was five parameters from 0.5 to 0.001. For all simulation numerical results, the constructed DNN has one input layer, three hidden layers, and one output layer, each hidden layer is with 200 neurons. The network input is a set of channel coefficient matrices. The channel coefficient matrix is a complex matrix. However, in order to facilitate the training of network, the modulus value of the channel coefficients will be used as input, so that the DNN works in the real domain. The size of the matrix depends on the number of users, the number of antennas and the number of antennas. The output of the network is the result of node control, the optimal beamforming matrix and the optimal power consumption. The beamforming matrix is usually a complex matrix, but in order to uniformly process in the real domain, this study took modulus values of each element of the matrix.

5.4 DNN Performance Evaluation

The evaluation was made in following three aspects: (1) the fit of beamforming matrix w generated by the DNN method and the traditional optimal algorithm, and the mean-square error of modulus value of the elements of two w were taken. If the sum of the mean-square error is smaller, showing the constructed DNN is more compatible with the traditional optimal algorithm; (2) For the total power generated by the two methods, the two powers are compared, if two power are almost the same, it proves that the constructed DNN can achieve optimal nodes management and beamforming; (3) When the two methods are used to solve the same problem, if the time spent on the constructed DNN is significantly less, then it is meaningful to apply it to solve the practical problem.

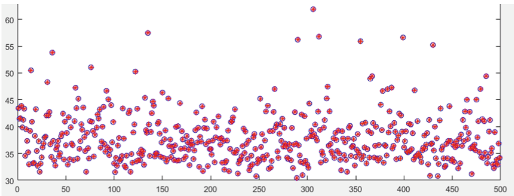
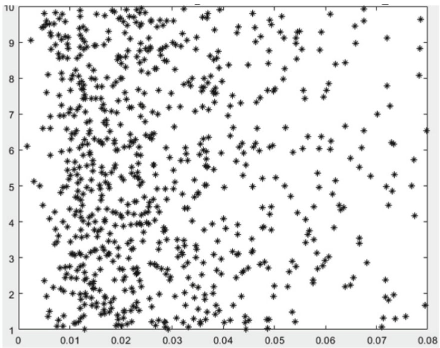


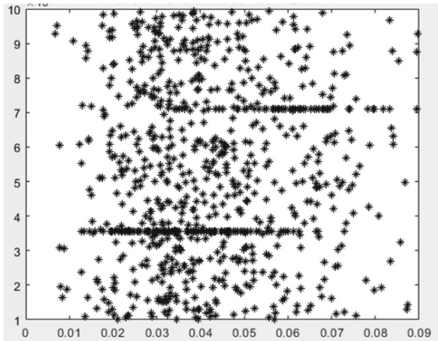
Fig. 3. Power scatter plot

Figure 3 shows the power consumption similarity obtained by the traditional algorithm and the DNN method when the number of antenna and UE are configured as (32,8). The x-axis is the channel realizations while the y-axis denotes the total consumed power. The circle and plus sign represent respectively total transmission power obtained from the traditional method and DNN method. It can be seen through 500 data points of the test set that the power consumption at each point can be well studied, and there are few points that have not been learned, indicating that the fit of the two methods is higher, and the compatible accuracy of the two is shown in Fig. 4.

In order to demonstrate the similarity of the two methods, the mean-square error and a joint scatter plot of the power consumption difference-beamforming matrix in Fig. 4 is used, where the configuration of antenna number and user number is set as (64,8), (128,16), respectively. It can be seen from the figure that accuracy of power



(64,8)



(128,16)

Fig. 4. Power difference and mean-square error joint scatter plot

consumption difference reaches the magnitude of 10^{-15} order, which can be completely coincident with almost every point learned; and the sum of the mean-square error of the elements of the beamforming matrix is at most 0.1, and the accuracy of most points is less than 0.05, proving every point can be learned very accurately.

6 Conclusion

In summary, DNN as a substitute for the traditional optimization algorithm which has a small amount of calculation and a short processing time. The DNN can precisely approximates the traditional algorithm, and the accuracy can reach the magnitude of 10^{-15}, achieving expected goal. This can well meet actual needs and significantly save energy consumption while satisfying user needs, thereby reducing operating costs to meet requirements of green communication. On the other hand, compared with the traditional numerical optimization method, the calculation process of DNN is low in complexity. This can significantly reduce calculation time, which is suitable for dealing with the complex combination optimization problems in dynamic scenes, thereby meeting needs of reality, having great guiding significance for solving real-time communication problems.

Acknowledgment. This work was supported by the Beijing City Board of education project (NO. KM201810858004).

References

1. Hoydis, J., Brink, S.T., Debbah, M.: Massive MIMO: how many antennas do we need? In: Communication, Control, & Computing (2011)
2. Larsson, E.G., Edfors, O., Tufvesson, F., et al.: Massive MIMO for next generation wireless systems. IEEE Commun. Mag. **52**(2), 186–195 (2014)
3. Ngo, H.Q., Ashikhmin, A., Hong, Y., et al.: Cell-free massive MIMO versus small cells. IEEE Trans. Wireless Commun. **16**(3), 1834–1850 (2017)
4. Wu, J., Zhang, Y., Zukerman, M., et al.: Energy-efficient base-stations sleep-mode techniques in green cellular networks: a survey. IEEE Commun. Surv. Tutorials **17**(2), 803–826 (2015)
5. Zhou, S., Gong, J., Yang, Z., et al.: Green mobile access network with dynamic base station energy saving. ACM MobiCom **9**(262), 10–12 (2009)
6. Conte, A., Feki, A., Chiaraviglio, L., et al.: Cell wilting and blossoming for energy efficiency. IEEE Wirel. Commun. **18**(5), 50–57 (2011)
7. Niu, Z.: TANGO: traffic-aware network planning and green operation. IEEE Wirel. Commun. **18**(5), 25–29 (2011)
8. Shi, Y., Zhang, J., Letaief, K.B.: Group sparse beamforming for green cloud-RAN. IEEE Trans. Wireless Commun. **13**(5), 2809–2823 (2014)
9. Dapeng, W., Yan, J., Wang, H., Dalei, W., Wang, R.: Social attribute aware incentive mechanism for device-to-device video distribution. IEEE Trans. Multimedia **19**(8), 1908–1920 (2017)
10. Zhang, H., et al.: Fronthauling for 5G LTE-U ultra dense cloud small cell networks. IEEE Wireless Commun. **23**(6), 48–53 (2017)
11. Zhang, H., et al.: Resource allocation for cognitive small cell networks: a cooperative bargaining game theoretic approach. IEEE Trans. Wireless Commun. **14**(6), 3481–3493 (2015)
12. Zhang, H., Huang, S., Jiang, C., et al.: Energy efficient user association and power allocation in millimeter wave based ultra dense networks with energy harvesting base stations. IEEE J. Sel. Areas Commun. **35**(9), 1936–1947 (2017)
13. Liang, S., Srikant, R.: Why deep neural networks for function approximation? In: ICLR, France Toulon, pp. 1–42 (2017)
14. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436 (2015)
15. Wu, D., et al.: Dynamic coding control in social intermittent connectivity wireless networks. IEEE Trans. Veh. Technol. **65**(9), 7634–7646 (2016)
16. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**(5), 359–366 (1989)