SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

August 2021, Vol. 64 182304:1–182304:10
https://doi.org/10.1007/s11432-020-2993-6

Deep learning based user scheduling for massive MIMO downlink system

Xiaoxiang YU, Jiajia GUO, Xiao LI* & Shi JIN

The National Communications Research Laboratory, Southeast University, Nanjing 210096, *China*

Received 28 March 2020/Revised 30 May 2020/Accepted 20 July 2020/Published online 1 June 2021

Abstract In this paper, we investigate a user scheduling algorithm for massive multiple-input multiple-output (MIMO) systems over more general correlated Rician fading channels. To achieve low latency and high throughput, a new user scheduling algorithm based on deep learning (DL) is proposed, which exploits only statistical channel state information. The proposed scheduling network is trained to grasp the mapping from the statistical signal and interference pattern to the user scheduling decision through supervised learning. It can predict the optimal scheduling scheme from statistical CSI without iterative calculation after offline training. Simulation results demonstrate the superior performance of the proposed algorithm in terms of calculation time, and it achieves almost the same throughput as the optimal scheduling algorithm which is obtained through exhaustive search. Furthermore, with the normalization of the input data, the proposed scheduling network is robust to the change of the channel environment and the number of transmit antennas.

Keywords massive MIMO, deep learning, statistical CSI, user scheduling

Citation Yu X X, Guo J J, Li X, et al. Deep learning based user scheduling for massive MIMO downlink system. Sci China Inf Sci, 2021, 64(8): 182304, https://doi.org/10.1007/s11432-020-2993-6

1 Introduction

Owing to its high throughput, massive multiple-input multiple-output (MIMO) has been recognized as one of the potential techniques for future wireless communication systems [1–3]. For multiuser systems, the availability of channel state information (CSI) at the base station (BS) is essential to the downlink scheduling so as to obtain the multiuser diversity gain. Although the downlink CSI can be estimated by exploiting the channel reciprocity from uplink training in time division duplexing (TDD) systems [4], it turns out that instantaneous CSI required at BS for massive MIMO systems in frequency division duplexing (FDD) mode leads to enormous training and feedback overhead. Moreover, performing scheduling for every channel realization leads to additional time overhead. It is unaffordable especially when the number of users is large and the users are moving rapidly. An alternative approach is to exploit the statistical CSI [5–7], which changes relatively slower than the instantaneous CSI. The statistical CSI can be obtained at BS through long-term feedback or even statistical reciprocity [8] between the transmitter and receiver. Therefore, the utilization of statistical CSI can greatly reduce the amount of feedback overhead as well as the scheduling complexity and frequency.

Recently, deep learning (DL) has attracted wide attention in wireless communication [9–12] due to its outstanding ability to deal with models that are hard to approximate or even unknown [13]. It has been successfully applied in many applications, such as CSI acquisition [10, 11], signal detection [14], and precoding [15]. For a densely deployed device-to-device network, Ref. [16] proposed a deep spatial learning approach to handle the optimal scheduling problem. Exploiting only statistical CSI, a DL-based beamforming algorithm was developed in [17], several scheduling algorithms were proposed through either exhaustive search or iteration algorithm which still incurs high time delay. Benefiting from the utilization of parallel processing and distributed memory architectures such as graphical processing units (GPUs),

* Corresponding author (email: li_xiao@seu.edu.cn)

the DL method manifests a remarkable potential to improve computational efficiency. Therefore, it is practical to use the DL method to reduce the computational delay of user scheduling.

Motivated by the above observation, we investigate the DL-based scheduling algorithm for massive MIMO systems over correlated Rician fading channels. In the offline training process, we construct a scheduling network to learn the mapping from the signal and interference pattern to the user scheduling decision. The optimal scheduling scheme of the system can be well predicted by the trained network based on statistical CSI with a short time delay. Moreover, the normalized input makes the trained scheduling network robust to the change of the channel environment and the number of transmit antennas.

The rest of this paper is organized as follows. In Section 2, the system model is introduced. In Section 3, a DL-based user scheduling algorithm is proposed and detailedly described. To validate the performance of the proposed DL-based user scheduling algorithm, simulation results are given in Section 4. Then, we conclude the paper in Section 5.

Notation. Vectors are represented as columns and are denoted in lower-case bold-faced characters, and matrices are represented in upper-case bold-faced. The superscript $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{\mathrm{H}}$ indicate the matrix transpose and conjugate-transpose operation, respectively. The complex number field is represented by $\mathbb{C}$, the real number field is represented by $\mathbb{R}$, $\mathrm{E}\{\cdot\}$ evaluates the expectation of the input, $\mathrm{Var}\{\cdot\}$ evaluates the variance of the input, and $\boldsymbol{I}_d$ denotes a $d \times d$ identity matrix.

2 System model

In this paper, a single-cell massive MIMO downlink transmission system is considered. The BS is equipped with uniform linear antenna array (ULA) with M antennas. We assume that there are L single-antenna users and at most U_t users can be served by BS. The received signal of user u is written as

$$y_u = \sqrt{p_u}\boldsymbol{h}_u^{\mathrm{T}}\boldsymbol{b}_u s_u + \sum_{i=1, i\neq u}^{U_t} \sqrt{p_i}\boldsymbol{h}_u^{\mathrm{T}}\boldsymbol{b}_i s_i + n_u, \tag{1}$$

where $\boldsymbol{h}_u \in \mathbb{C}^{M\times 1}$ represents the channel vector between the BS and user u, $s_i \sim \mathcal{CN}(0,1)$ is the transmitted symbol for user i, $\boldsymbol{b}_i \in \mathbb{C}^{M\times 1}$ is the unit-norm beamforming vector of user i, $n_u \sim \mathcal{CN}(0,\sigma_u^2)$ represents the complex additive white Gaussian noise. We assume that $p_i = P_t/U_t$ with P_t be the total transmitted power, which means equal power is assigned among the scheduled users.

In this paper, we consider the correlated Rician fading channel model, which is a relatively general model that includes both Rayleigh fading and uncorrelated Rician fading as its special cases. This model captures both the effect of line-of-sight (LoS) path and antenna correlation, since in a realistic situation the antennas might not be sufficiently separated and the propagation environment might not provide a sufficient amount of rich scattering [18, 19]. Under the correlated Rician fading channel model, the channel vector $\boldsymbol{h}_u$ is expressed as [17]

$$\boldsymbol{h}_u^{\mathrm{T}} = \sqrt{\frac{K_u}{K_u+1}}\bar{\boldsymbol{h}}_u^{\mathrm{T}} + \sqrt{\frac{1}{K_u+1}}\boldsymbol{h}_{w,u}^{\mathrm{T}}\boldsymbol{R}_u^{1/2}, \tag{2}$$

where $\bar{\boldsymbol{h}}_u \in \mathbb{C}^{M\times 1}$ is the LoS component, $\boldsymbol{h}_{w,u} \in \mathbb{C}^{M\times 1}$ contains independent and identically distributed (i.i.d.) complex Gaussian random variables with zero mean and unit variance. $\boldsymbol{R}_u \in \mathbb{C}^{M\times M}$ is the channel correlation matrix and has unit diagonal elements [6]. For ULA at the BS, the LoS component $\bar{\boldsymbol{h}}_u$ is given by [20]

$$\bar{\boldsymbol{h}}_u = \left[1, \mathrm{e}^{-\mathrm{j}\frac{2\pi d}{\lambda}\sin\theta_u}, \ldots, \mathrm{e}^{-\mathrm{j}\frac{2\pi d}{\lambda}(M-1)\sin\theta_u}\right]^{\mathrm{T}}, \tag{3}$$

where d is the distance between adjacent antennas, λ is the wavelength, θ_u is the angle-of-departure (AoD) of user u. Assume that each user has perfect CSI of its own and the BS merely has the statistical CSI of all users, i.e., K_u, σ_u^2, $\bar{\boldsymbol{h}}_u$ and $\boldsymbol{R}_u$.

From [17], the optimal beamforming vector exploiting only statistical CSI is obtained to be the $\bar{m}_u$-th column of the $M \times M$ discrete Fourier transform (DFT) matrix $\boldsymbol{F}_M$, i.e.,

$$\boldsymbol{b}_u = (\boldsymbol{F}_M)_{\bar{m}_u}, \tag{4}$$

where $\bar{m}_u$ is the index of the maximum diagonal element of $\boldsymbol{\Omega}_u$, which is given by

$$\boldsymbol{\Omega}_u = \boldsymbol{F}_M^{\mathrm{H}} \left(\frac{1}{K_u+1} \boldsymbol{R}_u + \frac{K_u}{K_u+1} \bar{\boldsymbol{h}}_u^* \bar{\boldsymbol{h}}_u^{\mathrm{T}} \right) \boldsymbol{F}_M. \tag{5}$$

Note that achieving high sum rate is usually an important consideration in transmission algorithm design. Thus, we investigate the user scheduling algorithm to maximize the ergodic sum rate under the beamforming vector (4). Let $q^{(u)} \in \{0,1\}$ be the scheduling indicator for user u, which equals to 1 if the user is scheduled, and 0 otherwise. For the considered system, the scheduling problem equals to determine the optimal $\boldsymbol{q} = [q^{(1)}, q^{(2)}, \ldots, q^{(L)}]^{\mathrm{T}}$ to maximize the ergodic sum rate, i.e.,

$$\max_{\boldsymbol{q}} \sum_{u=1}^{L} R_u, \tag{6a}$$

$$\begin{aligned} \text{s.t.} \quad & q^{(u)} \in \{0,1\}, \quad \forall u \in L, \\ & q^{(1)} + q^{(2)} + \cdots + q^{(L)} = U_t, \end{aligned} \tag{6b}$$

where R_u is the ergodic rate of user u calculated as

$$R_u = \mathrm{E}\left\{ \log_2 \left(1 + \frac{\frac{P_t}{U_t} |\boldsymbol{h}_u^{\mathrm{T}} \boldsymbol{b}_u|^2 q^{(u)}}{\sigma_u^2 + \frac{P_t}{U_t} \sum_{i=1, i\neq u}^{L} |\boldsymbol{h}_u^{\mathrm{T}} \boldsymbol{b}_i|^2 q^{(i)}} \right) \right\}. \tag{7}$$

3 DL-based user scheduling algorithm

Note that optimizing (6) directly is a challenging problem since there lacks tractable analytical expression of (7) under the correlated Rician fading channel. To solve this problem, a tractable approximation of the ergodic user rate under the beamforming vector (4) was derived in [17] and was shown to match well with the Monte-Carlo result. Based on it, an exhaustive search scheduling algorithm, as well as some reduced-complexity iteration algorithms was proposed. However, there still exists a computational bottleneck to achieve the optimal user scheduling algorithm in real-time. In this paper, we propose a user scheduling network through the DL method to reduce the latency of the scheduling algorithm.

3.1 Network input: signal and interference pattern

From [17], the ergodic rate of user u in (7) can be approximated as

$$R_u \approx \hat{R}_u = \log_2 \left(\frac{1 + \frac{\rho_u}{U_t} \sum_{i=1}^{L} \omega_u^{(\bar{m}_i)} q^{(i)}}{1 + \frac{\rho_u}{U_t} \sum_{i=1, i\neq u}^{L} \omega_u^{(\bar{m}_i)} q^{(i)}} \right), \tag{8}$$

where $\rho_u = P_t/\sigma_u^2$, $\omega_u^{(\bar{m}_u)}$ is the $\bar{m}_u$-th diagonal element of $\boldsymbol{\Omega}_u$ denoting the signal power in the optimal transmission direction, $\omega_u^{(\bar{m}_i)}$, $i \neq u$ reveals the interference power received from the user i.

The approximation of the ergodic rate (8) indicates that whether user u should be scheduled depends on the signal power $\omega_u^{(\bar{m}_u)}$ and the interference power $\omega_u^{(\bar{m}_i)}$. Thus, the input of the network should be the signal and interference pattern of the system. Moreover, the max-min normalization can be performed on the input data, which maps each element of the input to $[0,1]$. Through this normalization, the output of the network is only affected by the relative quantity of the signal and interference, which makes the network robust to different environments. The normalized input $\boldsymbol{X}_{\mathrm{in}} \in \mathbb{R}^{L\times L}$ is given by

$$\boldsymbol{X}_{\mathrm{in}} = \begin{pmatrix} x_1^{(\bar{m}_1)} & x_1^{(\bar{m}_2)} & \cdots & x_1^{(\bar{m}_L)} \\ x_2^{(\bar{m}_1)} & x_2^{(\bar{m}_2)} & \cdots & x_2^{(\bar{m}_L)} \\ \vdots & \vdots & & \vdots \\ x_L^{(\bar{m}_1)} & x_L^{(\bar{m}_2)} & \cdots & x_L^{(\bar{m}_L)} \end{pmatrix}, \tag{9}$$

where

$$x_i^{(\bar{m}_j)} = \frac{\omega_i^{(\bar{m}_j)} - \omega_{\min}}{\omega_{\max} - \omega_{\min}}, \tag{10}$$

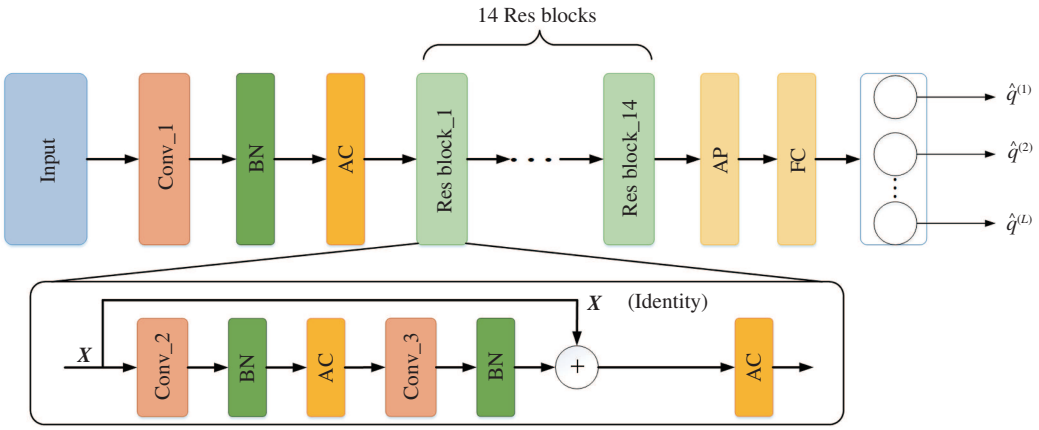

Figure 1 (Color online) The overall structure of the proposed scheduling neural network.

$$\omega_{\max} = \max_{1\leqslant i\leqslant L, 1\leqslant j\leqslant L} \omega_i^{(\bar{m}_j)}, \tag{11}$$

and

$$\omega_{\min} = \min_{1\leqslant i\leqslant L, 1\leqslant j\leqslant L} \omega_i^{(\bar{m}_j)}. \tag{12}$$

3.2 Network structure

The proposed scheduling network is based on the convolutional neural network and deep residual learning, as shown in Figure 1. The network includes a convolution (Conv) layer, a batch normalization (BN) layer, an activation (AC) layer, a residual block, an average pool (AP), and a fully connected (FC) layer[1)]. The convolutional neural network uses multiple processing layers to learn high-level features in training data, and exploits the signal correlation using local connections, shared weights and pooling, thus playing an important role in classification.

3.2.1 *Conv layer*

In the Conv layer, different filters with smaller dimensions and the same channels as the input are used to detect the presence of specific features in the original input, such as the signal and the interference pattern of each user. The filter convolves over the whole input matrix starting at the top left corner, computing the dot product of the pixel value in the original input and the values in the filter to generate a feature map. In fact, each element of a feature map is connected with the local data in the previous layer through the convolution kernel. Let $\boldsymbol{X} \in \mathbb{R}^{M_1\times N_1\times C}$ represent the input of Conv layer, $\boldsymbol{W}_k \in \mathbb{R}^{A\times B\times C}, k = 1,\ldots,K$ be the k-th convolutional kernel, $\boldsymbol{Y}_k \in \mathbb{R}^{M_1'\times N_1'}$, $k = 1,\ldots,K$ denote the k-th feature map generated by Conv layer, $M_1' = \lfloor \frac{M_1-A+2}{s} \rfloor + 1$, $N_1' = \lfloor \frac{N_1-B+2}{s} \rfloor + 1$, $s \geqslant 1$ be an integer parameter called stride, and K be the number of convolutional kernel. Then the (i,j)-th element of $\boldsymbol{Y}_k$, i.e., $y_k^{(i,j)}$, is calculated as

$$y_k^{(i,j)} = \sum_{a=1}^{A}\sum_{b=1}^{B}\sum_{c=1}^{C} x^{(si-s+a-1,sj-s+b-1,c)} w_k^{(a,b,c)} + \beta_k, \tag{13}$$

where $x^{(m_1,n_1,c)}$ is the (m_1,n_1,c)-th element of $\boldsymbol{X}$, $x^{(m_1,n_1,c)} = 0$ for all $m_1 \notin [1,M_1]$ and $n_1 \notin [1,N_1]$, $w_k^{(a,b,c)}$ is the (a,b,c)-th element of $\boldsymbol{W}_k$, and β_k is the bias of feature map k.

3.2.2 *BN layer and AC layer*

The role of the BN layer is to make the input of each layer of the neural network to have the same distribution during the training process, so as to solve the problem of difficult training and slow convergence

1) The design of this deep neural network is to solve the user scheduling problem, although the input dimension might not be large, as shown in simulation, the essence of the problem is quite different from image processing. The scheduling network needs to find one optimal scheduling solution from a large number of possible solutions. Due to the complexity of the problem, it needs a large number of training samples (as shown in the simulation results) to contain all the possible scheduling schemes and a relatively deep neural network to ensure the accuracy of the prediction result.

caused by the deepening of the network. In the proposed network, the BN layer normalizes the output of the Conv layer by subtracting the batch expectation and dividing by the batch standard deviation [21]. For a layer with k-dimensional input $\boldsymbol{c} = (c^{(1)}, \ldots, c^{(k)})$, each dimension will be normalized by

$$\hat{c}^{(i)} = \frac{c^{(i)} - \mathrm{E}[c^{(i)}]}{\sqrt{\mathrm{Var}[c^{(i)}]}}, \tag{14}$$

where the expectation and variance are computed over the training dataset. However, this simple normalization may change what the layer can represent. To solve this problem, two trainable variables $\gamma^{(i)}$ and $\beta^{(i)}$ are introduced to scale and shift the normalized value:

$$z_{\mathrm{BN}}^{(i)} = \gamma^{(i)}\hat{c}^{(i)} + \beta^{(i)}, \tag{15}$$

where $\gamma^{(i)}$ and $\beta^{(i)}$ are learned during the training process to restore the presentation capability.

The AC layer is the follow-up of BN layer which is used to increase the non-linear expression ability of the neural network. For the intermediate AC layers, the rectified linear unit (ReLU) with the expression $f_{\mathrm{ReLU}}(x) = \max(x, 0)$ is selected as the activation function.

3.2.3 *Residual block*

In the field of DL, the number of layers is generally considered to be a key factor that affects the expression ability of the neural network. However, as the number of layers increases, the notorious problem of gradient explosion often occurs, which leads to performance degradation. To address this issue, we apply a deep residual network (ResNet) [22]. In Figure 1, there is a shortcut connection every two Conv layers, so that the neural network only needs to input $\boldsymbol{X}$, and mapping the difference between the desired underlying mapping $H(\boldsymbol{X})$ and $\boldsymbol{X}$ itself, i.e., $F(\boldsymbol{X}) = H(\boldsymbol{X}) - \boldsymbol{X}$, thus greatly reduces the difficulty of learning.

3.2.4 *AP layer and FC layer*

The high-dimensional features are extracted from initial input through multiple Conv layers. Then an AP layer is used to simplify the network and select the main features. Let $\boldsymbol{X}_i$ denote the i-th input feature map of the AP layer, then the i-th output $z_{\mathrm{AP}}^{(i)}$ is given by the average value of all elements in $\boldsymbol{X}_i$.

The FC layer serves as the interpret module of the features extracted from the previous layers, and each output neuron of this layer applies ReLU activation function to a weighted sum of the output of AP layer, thus the i-th output is

$$z_{\mathrm{FC}}^{(i)} = f_{\mathrm{ReLU}}\left(\sum_j w_{\mathrm{FC}}^{(i,j)} z_{\mathrm{AP}}^{(j)} + \beta_{\mathrm{FC}}^{(i)}\right), \tag{16}$$

where $w_{\mathrm{FC}}^{(i,j)}$ is the weight of the j-th output of AP layer connected to the i-th neuron of FC layer, $\beta_{\mathrm{FC}}^{(i)}$ is the bias of this layer.

3.2.5 *Scheduling outputs*

A general DL method selects one out of the overall result value combinations for the label, which is called a multi-class classification problem. Theoretically, for the case of user scheduling, the category of output label is $\frac{L!}{U_t!(L-U_t)!}$, corresponding to all possible schemes. Although it is possible to adopt a deep neural network as well as a large number of training samples to ensure the predictive ability of the neural network, the output dimension of the neural network is not practical to deploy when the number of users grows large. To address this issue, we treat this problem as multi-label classification which applies the sigmoid activation function in the output layer. The i-th neuron output

$$\hat{q}^{(i)} = \left(1 + \mathrm{e}^{-(\sum_j w_{\mathrm{out}}^{(i,j)} z_{\mathrm{FC}}^{(j)} + \beta_{\mathrm{out}}^{(i)})}\right)^{-1}, \tag{17}$$

where $w_{\mathrm{out}}^{(i,j)}$ and $\beta_{\mathrm{out}}^{(i)}$ are the weights and bias of this layer. Thus, each output $\hat{q}^{(i)}$ is a value between 0 and 1, indicating the probability of the corresponding user that will be served by the BS, the size of the output neurons is reduced to L. Finally, we select the U_t users from $\hat{q}^{(i)}$, $i = 1, \ldots L$ with the highest probability value as the scheduling result.

3.3 Offline training and online testing

Note that supervised learning is utilized to train the proposed network, thus the loss function is crucial for the training process. Since the output neuron of the network indicates the probability of scheduling the corresponding user, the cross-entropy (CE) loss function is suitable to describe the difference between the labels and predictions of the network. Moreover, an $l2$-norm regularization term is added to the loss function to avoid overfitting. Then the loss function applied in this algorithm is expressed as

$$J(\boldsymbol{\theta}) = \varepsilon\|\boldsymbol{w}\|_2^2 - \frac{1}{N}\sum_{n=1}^{N}\sum_{l=1}^{L}\left[q_n^{(l)}\ln(\hat{q}_n^{(l)}) + (1-q_n^{(l)})\ln(1-\hat{q}_n^{(l)})\right], \tag{18}$$

where N is the number of training samples, ε is the regularization coefficient, $\boldsymbol{w}$ is the weights of the entire network, $q_n^{(l)}$ is the training label representing the scheduling result of the l-th user in the n-th sample, $\hat{q}_n^{(l)}$ is the scheduling probability of the corresponding l-th user predicted by the network. The parameters $\boldsymbol{\theta}$ (all the weights, bias and learnable parameters of BN layers) of the neural network are updated by minimizing the loss function during the training process. The training labels are generated by maximizing the following ergodic sum rate approximation through exhaustive search:

$$\max_{\boldsymbol{q}} \sum_{u=1}^{L} \hat{R}_u, \tag{19a}$$

$$\begin{aligned} \text{s.t.} \quad & q^{(u)} \in \{0,1\}, \quad \forall u \in L, \\ & q^{(1)} + q^{(2)} + \cdots + q^{(L)} = U_t, \end{aligned} \tag{19b}$$

which we referred to as the "sum rate based" algorithm.

In the online prediction, the BS only needs to obtain the normalized signal and interference pattern $\boldsymbol{X}_{\text{in}}$, and inputs it to the network to get the corresponding scheduling scheme.

4 Simulation

In this section, we carry out the simulation results to evaluate the performance of the proposed scheduling network. The performance of two benchmark algorithms are compared, i.e., the "most dissimilar" algorithm proposed in [17] and the "sum rate based" scheduling algorithm that maximizes the ergodic sum rate approximation through exhaustive search. We first consider a massive MIMO downlink transmission system with $L = 20$, $M = 64$, $U_t = 6$. Assume that the noise level $\sigma_u^2 = \sigma^2$ and the SNR is defined as P_t/σ^2. We randomly generate 3.1×10^6 samples of user drops. In each drop, the AoD of each user is uniformly distribution in $(-90°, 90°)$, the angle spread (AS) is uniformly distributed in $(5°, 15°)$, the Rician K-factor is uniformly distributed in $[-10, 10]$ dB, and the one-ring scattering model [6] is adopted to determine the channel correlation matrix $\boldsymbol{R}_u$. Then, assuming SNR = 10 dB, we use the sum rate based algorithm to generate the label of training sample for each specific user drops, and the above samples are divided into training set and validation set that contains 3×10^6 and 10^5 samples respectively. The layout of the network is shown in Table 1, the ReLU activation function is used in all Res block and down-sampling is performed by the first Conv layer of Res block 1 and 13 with a stride of 2.

In the offline training process, a single GPU of NVIDIA RTX2080 Ti is applied to train the scheduling network. We use the stochastic gradient descent optimizer to minimize the loss function and settle the parameters of the network with the highest prediction accuracy on the validation set. The batch size is 1024, the training epoch is 150, and the regularization coefficient ε is 0.0001. We set the initial learning rate to be 0.1 and multiply it by 0.1 every 40 epochs.

Figure 2 shows the average sum rate performance of the proposed scheduling network as well as the two benchmark algorithms. It can be seen that the sum rate based algorithm achieves the highest average sum rate, since it is the optimal scheduling algorithm, which is used to generate the training labels for the proposed scheduling network. Although the average sum rate performance of the proposed scheduling network can not exceed the optimal scheduling algorithm, it can be observed that the average sum rate performance of the proposed scheduling network is almost the same as the optimal scheduling algorithm and outperforms the "most dissimilar" algorithm in the low and middle SNR regions. In the high SNR region, the proposed scheduling network is slightly inferior to the two benchmark algorithms. This is

Table 1 Layout of the scheduling network

Layer name	Output size	Convolution kernel
Conv1,BN,ReLU	$20 \times 20 \times 32$	$3 \times 3, 32$
Res block: 1–2	$10 \times 10 \times 32$	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 2$
Res block: 3–12	$10 \times 10 \times 64$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 10$
Res block: 13–14	$5 \times 5 \times 128$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$
AP	1×128	–
512-d FC	1×512	–
20-d sigmoid	1×20	–

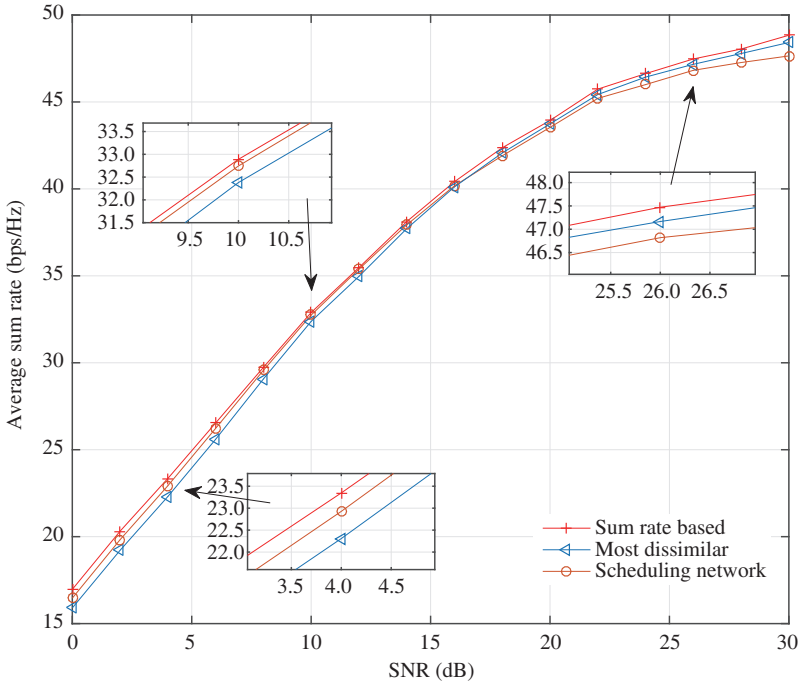

Figure 2 (Color online) Average sum rate comparison.

because all training data is generated under an SNR of 10 dB, and the network does not have any samples for the high SNR region. In another aspect, the system is interference limited in the high SNR region, while it is noise limited in the low SNR region. Thus, the optimal scheduling solutions are different under low and high SNR region. To retrieve the performance loss in the high SNR region, we trained another user scheduling network whose training data is generated under an SNR of 26 dB to predict user scheduling scheme when the SNR is higher than 20 dB. The performance of the proposed algorithm with two networks at different SNR is shown in Figure 3. It can be seen that the performance of the proposed algorithm is improved in the high SNR region, and is almost the same as the optimal algorithm.

Table 2 compares the normalized computation time of these scheduling algorithms in Figure 2. All computation time is normalized by the time consumption of the proposed scheduling network. Because the scheduling network greatly exploits the parallel computing power of GPU, and the other two scheduling algorithms require sequential iteration, the time consumptions of the sum rate based and most dissimilar algorithms are more than 18.7 and 12.9 times of the proposed network, respectively. It can be clearly seen that the scheduling network greatly reduces the computational time required to obtain the scheduling scheme while ensuring the high spectral efficiency of the system.

Figures 4 and 5 verify the robustness of the proposed framework to the change of channel environment and the number of transmit antennas. In these two figures, the proposed DL-based algorithm only uses a single network trained by the dataset generated under SNR = 10 dB. In Figure 4, the simulation

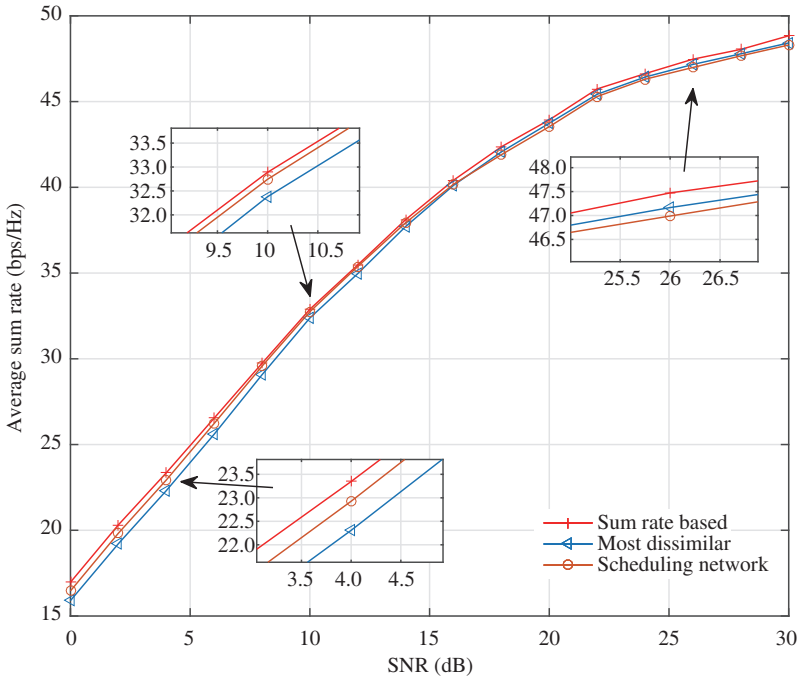

Figure 3 (Color online) Average sum rate comparison (with two trained networks).

Table 2 Comparison of normalized computation time

Scheduling algorithm	Normalized runtime
Sum rate based	18.7
Most dissimilar	12.9
Scheduling network	1

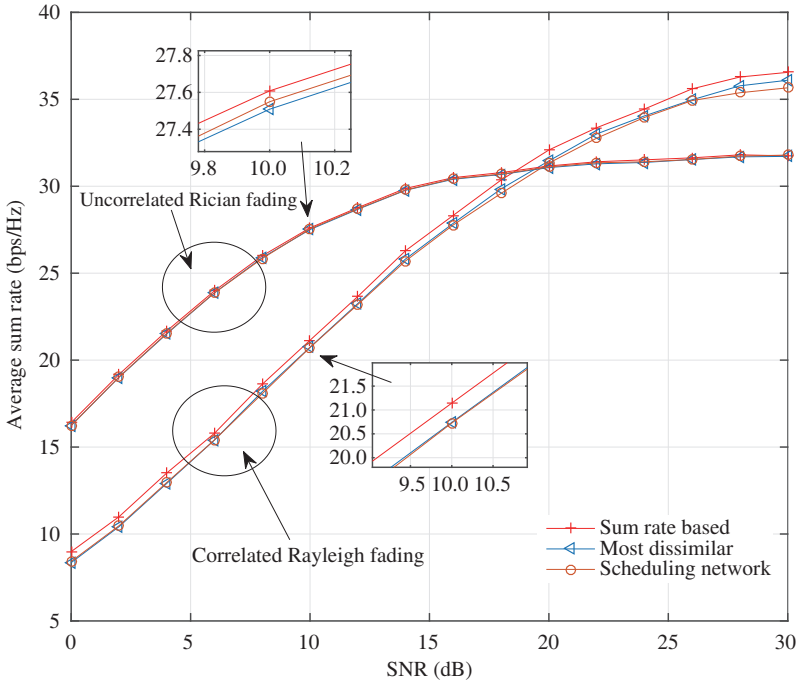

Figure 4 (Color online) Average sum rate comparison under different channel environments.

results under two channel environments are compared, i.e., the correlated Rayleigh fading ($K_u = 0$) and uncorrelated Rician fading ($\boldsymbol{R}_u = \boldsymbol{I}_M$). In Figure 5, the simulation results under two different numbers

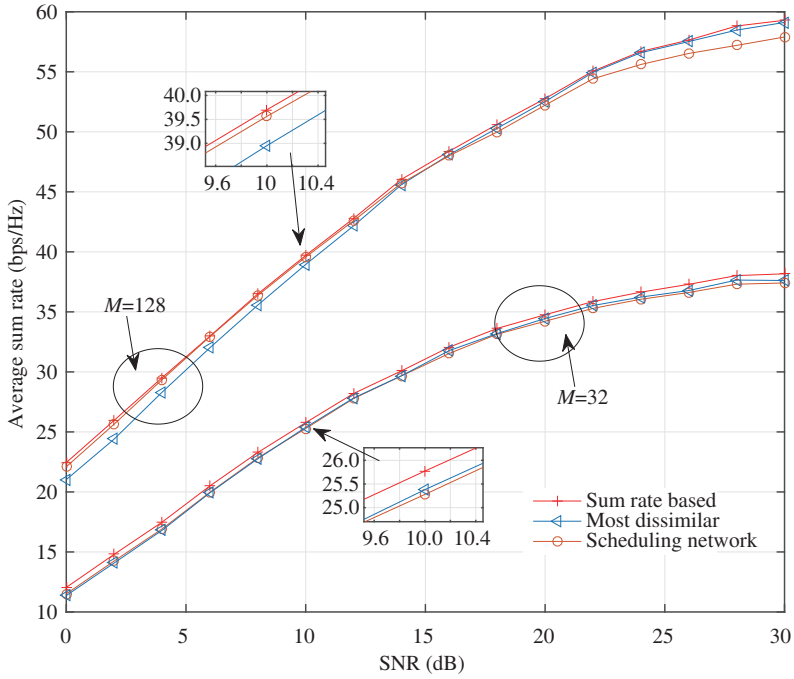

Figure 5 (Color online) Average sum rate comparison under different number of antenna.

of antenna, i.e., $M = 32$ and $M = 128$, are given. The other parameters are the same as Figure 2. From Figure 4, it can be seen that the proposed scheduling network performs almost the same with the optimal scheduling algorithm, i.e., the sum rate based algorithm, under the uncorrelated Rician fading channel, while its performance is very close to optimal under the correlated Rayleigh fading channel. From Figure 5, we can also observe that the performance gap between the proposed scheduling network and the sum rate based algorithm is almost negligible. These results indicate that the proposed scheduling network is robust to the change of the environment and the number of transmit antennas.

5 Conclusion

In this paper, we proposed a DL-based scheduling network for massive MIMO downlink systems with only statistical CSI known at the BS. The proposed scheduling network maps the statistical signal and interference pattern to the user scheduling decision through supervised learning. It can achieve almost the same performance as the optimal exhaustive search algorithm while having much lower latency. Moreover, the trained network can adapt to the change of the channel environment and the number of transmit antennas.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61831013, 61971126, 61941104, 61921004).

References

1 Rusek F, Persson D, Lau B K, et al. Scaling up MIMO: opportunities and challenges with very large arrays. IEEE Signal Process Mag, 2013, 30: 40–60
2 You X H, Pan Z W, Gao X Q, et al. The 5G mobile communication: The development trends and its emerging key techniques (in Chinese). Sci Sin Inform, 2014, 44: 551–563
3 Wang D M, Zhang Y, Wei H, et al. An overview of transmission theory and techniques of large-scale antenna systems for 5G wireless communications. Sci China Inf Sci, 2016, 59: 081301
4 Marzetta T L. Noncooperative cellular wireless with unlimited numbers of base station antennas. IEEE Trans Wirel Commun, 2010, 9: 3590–3600
5 Li X, Liu Z Y, Qin N N, et al. FFR based joint 3D beamforming interference coordination for multi-cell FD-MIMO downlink transmission systems. IEEE Trans Veh Technol, 2020, 69: 3105–3118
6 Adhikary A, Nam J, Ahn J-Y, et al. Joint spatial division and multiplexing: the large-scale array regime. IEEE Trans Inform Theory, 2013, 59: 6441–6463
7 Li X, Sun T T, Qin N N, et al. User scheduling for downlink FD-MIMO systems under Rician fading exploiting statistical CSI. Sci China Inf Sci, 2018, 61: 082302
8 Han Y, Hsu T H, Wen C K, et al. Efficient downlink channel reconstruction for FDD multi-antenna systems. IEEE Trans Wirel Commun, 2019, 18: 3161–3176

9 Wang T Q, Wen C K, Wang H Q, et al. Deep learning for wireless physical layer: opportunities and challenges. China Commun, 2017, 14: 92–111
10 Wen C K, Shih W T, Jin S. Deep learning for massive MIMO CSI feedback. IEEE Wirel Commun Lett, 2018, 7: 748–751
11 He H T, Wen C K, Jin S, et al. Deep learning-based channel estimation for beamspace mmWave massive MIMO systems. IEEE Wirel Commun Lett, 2018, 7: 852–855
12 He H T, Jin S, Wen C K, et al. Model-driven deep learning for physical layer communications. IEEE Wirel Commun, 2019, 26: 77–83
13 Goodfellow I, Bengio Y I, Courville A. Deep Learning. Cambridge: MIT Press, 2016
14 Ye H, Li G Y, Juang B H. Power of deep learning for channel estimation and signal detection in OFDM systems. IEEE Wirel Commun Lett, 2018, 7: 114–117
15 Xia W C, Zheng G, Zhu Y X, et al. A deep learning framework for optimization of MISO downlink beamforming. IEEE Trans Commun, 2020, 68: 1866–1880
16 Cui W, Shen K M, Yu W. Spatial deep learning for wireless scheduling. IEEE J Sel Areas Commun, 2019, 37: 1248–1261
17 Li X, Yu X X, Sun T T, et al. Joint scheduling and deep learning-based beamforming for FD-MIMO systems over correlated Rician fading. IEEE Access, 2019, 7: 118297–118309
18 Taricco G. Optimum receiver design and performance analysis of arbitrarily correlated Rician fading MIMO channels with imperfect channel state information. IEEE Trans Inform Theory, 2010, 56: 1114–1134
19 Boukhedimi I, Kammoun A, Alouini M S. Multi-cell MMSE combining over correlated Rician channels in massive MIMO systems. IEEE Wirel Commun Lett, 2020, 9: 12–16
20 Han Y, Zhang H C, Jin S, et al. Investigation of transmission schemes for millimeter-wave massive MU-MIMO systems. IEEE Syst J, 2017, 11: 72–83
21 Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of International Conference on Machine Learning, Lille, 2015. 448–456
22 He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 2016. 770–778