Machine Learning-Based Power Loading for Massive Parallel Gaussian Channels

Min Jeong Kang and Jung Hoon Lee
Department of Electronics Engineering and Applied Communications Research Center,
Hankuk University of Foreign Studies, Yongin, Korea
{love_minmin926, tantheta}@hufs.ac.kr

***Abstract*—In parallel channels, it is well known that the waterfilling is optimal power allocation that maximizes the sum achievable rate. However, the waterfilling requires iterative calculations, so may not be suitable especially when the number of total channels is very large or the delay constraint is very tight. In this paper, we propose machine learning-based power loading to reduce the computational complexity of power allocation in massive Gaussian parallel channels, which emulates on-off power allocation, where some of the channels equally share total transmit power. Our proposed scheme adopts a deep neural network structure that takes channel gains with total transmit power and returns an on-off power allocation strategy. The numerical results show that our proposed scheme achieves almost the same performance with the on-off power allocation with reduced complexity.**

***Index Terms*—Machine learning (ML), waterfilling, on-off power allocation, parallel channels, deep neural network (DNN).**

I. Introduction

With the development of machine learning technology, there have been many attempts to use machine learning for wireless communications. Machine learning can be applied for many purposes; one purpose is to solve difficult problems such as joint optimization, and another purpose is to reduce the computational complexity when the optimal solution can be found, but its complexity is prohibitive. One popular machine learning technology is deep neural network (DNN), which accelerated the development of machine learning in many research areas [1].

In parallel channels, it is well known that the optimal power allocation is waterfilling. However, the waterfilling iteratively finds optimal powers, so may not be suitable when the number of total subchannels is very large or the delay constraint is very tight. One suboptimal power allocation with reduced complexity is on-off power configuration [2], where some of channels (“on” channels) equally share total power budget, while the others (“off” channels) are turned off. Also, the authors of [3] proposed multi-level power loading that generalizes on-off power configuration. However, these power allocation schemes still requires high complexity when the number of total channels are very large. The authors of [4] proposed the waterfilling with reduced complexity, where the waterfilling is over the subchannel groups, and subchannels in the same group use the equal power.

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2021R1H1A1010858) and by Hankuk University of Foreign Studies Research Fund of 2021.

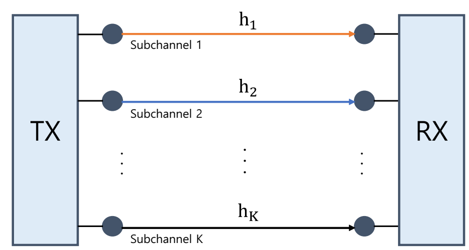

Fig. 1. System model.

In this paper, we propose machine learning-based power loading to reduce the computational complexity of power allocation in massive Gaussian parallel channels. Our proposed scheme adopts a deep neural network (DNN) and emulates on-off power allocation. Thus, once trained in offline, our proposed scheme (with the DNN structure) can find power loading sequences after finite clocks. We evaluate our proposed scheme and show that our proposed scheme achieves almost the same performance with the on-off power allocation with reduced complexity.

II. Problem Formulation

A. System Model

Our system model is illustrated in Fig.1. We consider a K-parallel Gaussian channel when the number of total subchannels (i.e., K) is very large. The received signal at the kth subchannel is modeled by

$$y_k = h_k \sqrt{p_k} s_k + n_k, \quad k \in \{1, ..., K\}, \tag{1}$$

where $h_k \in \mathbb{C}^{1\times 1}$ is the channel at the kth subchannel, which is a circularly symmetric complex Gaussian random variables with zero mean and unit variance,i.e.,$h_k \sim \mathcal{CN}(0, 1)$. In this paper, without loss of generality, we assume that the subchannel gains are sorted in descending order such that

$$|h_1|^2 \geq ... \geq |h_K|^2. \tag{2}$$

The variable $s_k \in \mathbb{C}^{1\times 1}$ is the kth transmit symbol such that $|s_k|^2 = 1$, and p_k is the transmit power for the kth symbol. Also, n_k is a complex Gaussian noise with zero mean and unit variance, i.e., $n_k \sim \mathcal{CN}(0,1)$. When total transmit power budget is P, it should be satisfied that

$$\sum_{k=1}^{K} p_k = P. \tag{3}$$

The achievable rate at the kth subchannel is

$$\mathcal{R}_k = \log_2\left(1 + p_k |h_k|^2\right), \tag{4}$$

so the sum achievable rate becomes $\sum_{k=1}^{K} \mathcal{R}_k$.

B. Waterfilling power allocation and on-off power configuration

In parallel Gaussian channels, it is well known that *the waterfilling* is optimal power allocation for sum rate maximization. With the waterfilling, the optimal power for the subchannel k is given by

$$p_k^\star = \left[\frac{1}{\mu} - \frac{1}{|h_k|^2}\right]^+, \tag{5}$$

where μ is a constant satisfying the total power constraint given in (3).

Although the waterfilling power allocation given in (5) maximizes the sum achievable rate, the value of μ should be calculated with an iterative manner, so the computational complexity becomes huge burden when the number of total subchannels (i.e., K) is very large or the delay constraint is very tight.

One suboptimal power allocation is on-off configuration [2], where only some of subchannels ("on" subchannels) equally share total power budget, while the others ("off" subchannels) are turned off. In this case, power allocation sequence $(p_1, \ldots, p_K)$ is one among K power allocation sequences given by

$$\left\{ \left([P/n]_n, [0]_{K-n}\right) \;\middle|\; n = 1, \ldots, K \right\}, \tag{6}$$

where $[m]_n$ denotes the sequence of consecutive 'm's repeated n times. For example, the power allocation $([P]_1, [0]_{K-1})$ indicates the total power allocation to the first subchannel, i.e., $([P]_1, [0]_{K-1}) = (P, 0, \ldots, 0)$, while $([P/N]_N, [0]_0)$ indicates equal power allocation over all subchannels, i.e., $([P/K]_K, [0]_0) = (P/K, \ldots, P/K)$.

When $n (\leq K)$ subchannels shares total powers, the achievable rate with on-off configuration, the achievable rate is given by

$$\mathcal{R}_{\text{sum}}^{\text{on-off}}(n) = \sum_{k=1}^{n} \log_2\left(1 + \frac{P}{n}|h_k|^2\right). \tag{7}$$

Thus, for the optimal on-off configuration, the transmitter need to compare total K power allocation sequences given in (6). The transmitter first solve the following problem

$$n^\star = \underset{n \in \{1, \ldots, K\}}{\arg\max}\; \mathcal{R}_{\text{sum}}^{\text{on-off}}(n), \tag{8}$$

and then obtain the optimal power allocation sequence as follows:

$$\left([P/n^\star]_{n^\star}, [0]_{K-n^\star}\right). \tag{9}$$

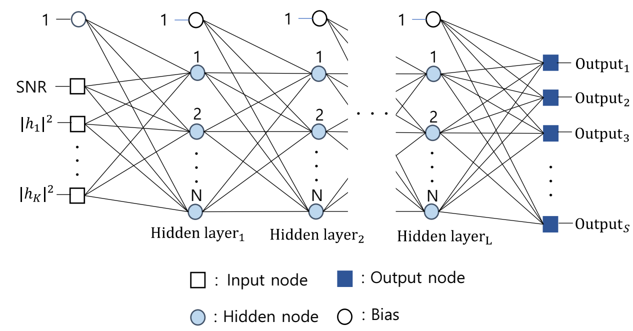

Fig. 2. Deep neural network (DNN) model for on-off power allocation.

III. Proposed Machine Learning-Based Power Loading

In this section, we explain our machine learning-based power loading scheme.

A. Basic idea

As we stated earlier, when waterfilling may not be suitable when the number of total subchannels is very large or the delay constraint is very tight. Although the suboptimal on-off configuration reduces the computational complexity of waterfilling, but still requires high complexity when the number of total subchannels are very large. Thus, to circumvent this complexity issue, we consider machine learning-based power loading that emulates the on-off power allocation (i.e., solves the problem in (8)). Note that once trained in offline, a DNN structure can yield a solution after finite clocks.

B. Structure of our proposed machine learning-based power loading

Fig. 2 shows the structure of our machine learning model. We consider a deep neural network (DNN) structure that takes the channel gains of K subchannels (i.e., $|h_1|^2, \ldots, |h_K|^2$) with total power budget (i.e., P) and returns the optimal number of turned-on subchannels (i.e., $n^\star$ in (8)). Thus, our machine learning structure has $K+1$ nodes in the input layer and K nodes in the output layer. Also, we consider L hidden layers, each of which is comprise of N nodes.

For activation functions, we employ the Rectified linear unit (ReLU) function and the Softmax function at each hidden node and each output node, respectively. Also, we consider the cross entropy for the loss function, which is widely used for weight correction in machine learning models defined as follows:

$$e = -\sum_{i=1}^{S} l_i \log_2 o_i + (1 - l_i) \log_2(1 - o_i), \tag{10}$$

where l_i is the ith label. Since only one of the s data labels is the correct answer, only the correct label has the value of one, and the rest ones have the value of zero. In (10), o_i is the i output data of the i output node. Also, for optimization, we employ the adaptive momentum (AdaM) algorithm. Overfitting may occur if the initial learning rate

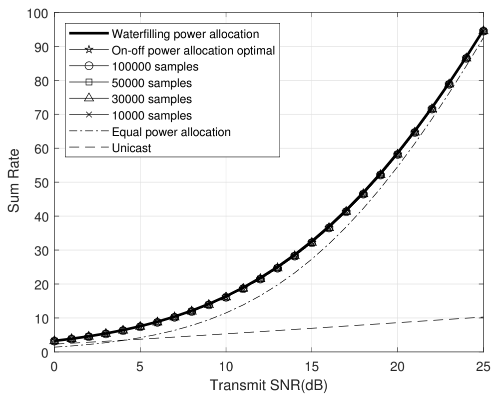

Fig. 3. The achievable sum rates of various schemes when $K = 32$.

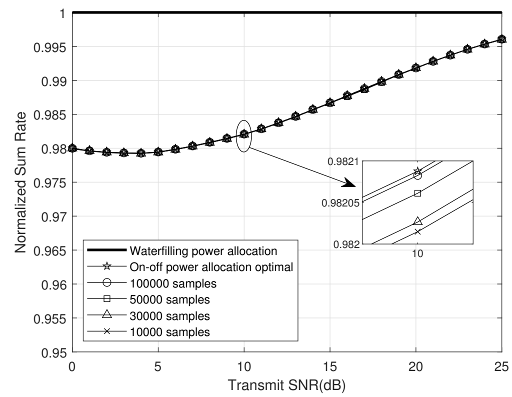

Fig. 4. The achievable sum rates of various schemes normalized with the achievable sum rate of the on-off configuration when $K = 32$.

is low, so the initial learning rate is set to 0.001. To prevent overfitting, we employed an early stop method, where training is stop when the loss value is not decreasing more than consecutive 20 times.

IV. Numerical Results

In this section, we evaluate our machine learning model. We assume that there are total 32 subchannels (i.e., $K = 32$). Thus, our machine learning model has 33 input nodes at the input layer and 32 output nodes at the output layer. Also, our machine learning model has five hidden layers, each of which has 600 nodes.

For evaluation, we vary the number of data samples when training our machine learning model. We consider four cases of $(1, 3, 5, 10) \times 10^4$ data samples and for each case, 80% of data samples is used for training and the remaining 20% is used for validation. Then, we measure the performances of the machine learning models with the same 5000 test data samples. As reference schemes, we consider the waterfilling power allocation, the on-off configuration, equal power allocation, and unicast to the best subchannel (i.e., whole power to the best subchannel).

Fig. 3 shows the achievable sum rates of various schemes, while Fig. 4 shows the normalized ones with the waterfilling. As we can see in Fig. 4, on-off power allocation achieves about or more than 98% performance compared to the waterfilling. Also, our proposed machine learning models achieve almost the same performance with on-off power allocation.

Fig.5 shows how accurate our machine learning models emulate the on-off power allocation. As we can see, the accuracy tends to increases as the number of sample data increases in four cases.

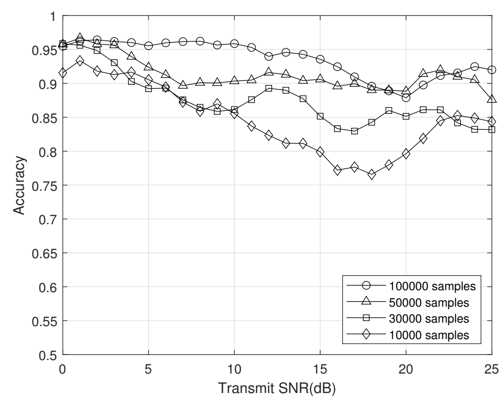

Fig. 5. The accuracy of our machine learning model when emulating the optimal on-off configuration.

V. Conclusion

In this paper, we proposed machine learning-based power loading to reduce the computational complexity of power allocation in massive Gaussian parallel channels, where the waterfilling may not be suitable because the number of total subchannels is very large or the delay constraint is very tight. Our proposed scheme emulates on-off power allocation, where some of subchannels equally share total transmit power. In numerical results, we showed that our proposed machine learning structure achieves almost the same performance with the on-off power allocation with reduced complexity.

Acknowledgment

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2021R1H1A1010858) and by Hankuk University of Foreign Studies Research Fund of 2021.

References

[1] J. Kaur, M. A. Khan, M. Iftikhar, M. Imran, and Q. E. U. Haq, "Machine learning techniques for 5G and beyond," *IEEE Access*, vol. 9, pp. 23472–23488, 2021.

[2] A. Leke and J. M. Cioffi, "A maximum rate loading algorithm for discrete multitone modulation systems," in Proc. *1997 IEEE Global Telecommunications Conf.*, vol. 3, pp. 1514–1518.

[3] J. H. Lee and W. Choi, "Multi-level power loading using limited feedback," *IEEE Commun. Lett.*, vol. 16, no. 12, pp. 2024–2027, Dec. 2012.

[4] Q. Qi, A. Minturn, and Y. Yang, "An efficient water-filling algorithm for power allocation in OFDM-based cognitive radio systems," in Proc. *2012 International Conference on Systems and Informatics*, pp. 2069–2073.